REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1997 (12/10/97)	Final Technical Report, June 1996 – August 1997

4. TITLE AND SUBTITLE

Clearning Phased Array Radar Data

5. FUNDING NUMBERS

6. AUTHOR(S)

Dr. Andreas S. Weigend

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

formerly University of Colorado at Boulder, now Leonard N. Stern School of Business, NYU

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

Air Force Office of Scientific Research
110 Duncan Avenue, Suite B115
Bolling AFB, DC 20332

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

a. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Unlimited

13. ABSTRACT (Maximum 200 words)

Many military and civilian problems can be viewed as pattern recognition: given a set of measured inputs, the task is to predict the corresponding output. Typical examples range from image recognition and classification, to time series prediction and regression. Most modeling assumes that the inputs can be measured exactly, without noise. Building a model then means to construct (or "learn") a mapping from these inputs to the expected values of the outputs. The usually tacit assumption of noise-free inputs is violated in most real-world problems where only a noisy version of the "true" input is observed.

This research found that while it was possible for *time series problems* even if there is a lot of noise present, to use information from adjacent patterns in time, the problem could be solved for non-time series problems, such as the phase array radar data.

The effort lead to several papers. Results are presented on discrete hidden states (Hidden Markov models), and continuous hidden state (state space models). A paper on finding the true inputs using Independent Component Analysis is in preparation. A paper on evaluation methodology using the bootstrap also employs the state space approach.

14. SUBJECT TERMS

Machine Learning. Errors in Variables. Orthogonal Distance Regression. Total Least Squares. Nonlinear Modeling. Time Series.

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Clearning Phased Array Radar
AFOSR Grant F49620-96-1-0240, Final Report
AFOSR Program Manager: Jon Sjogren

Principal Investigator: Andreas Weigend

Until December 1996: Assistant Professor, Computer Science Department, University of Colorado at Boulder

Since January 1997: Associate Professor, Information Systems Department, Leonard N. Stern School of Business, New York University

1. Objective

Observations, such as phased array radar data, contain noise, usually from several sources. The essence of modeling, and subsequent inference, is to extract the signal. The objective of this grant was to understand the strengths and limitation of a new algorithm called "clearning" (the combination of learning the model and cleaning the data), and to apply it to phased array radar data.

2. Results

The proposal was written with a three-year time horizon. Before applying the algorithm to phased array radar data, and comparing it to competing algorithms, the first goal was to understand what the algorithm can do, and what it cannot do. This was best done by relating it to the important research question: to what degree can we infer hidden states from observed data?

The key result is: hidden states can be inferred successfully for time series data. Time series data have the major advantage that adjacent patterns are indeed related to each other. This is not the case in standard, non-time-series pattern recognition problems.

The first progress report emphasized the important of constraints between the input variables to exist for clearning to work. In particular, it emphasized that the first steps of the project thus are to clarify what might be done, and what cannot be done in principle, as well as to relate clearning to source separation, and, in the case of time series, to state space modeling and Kalman filtering. This has been achieved: The following describes the research that my collaborators and I carried out in the last year in the context of finding ("hidden") variables (continuous, as in clearning, or discrete) that are a less noisy characterization of the systems than a snapshot of the raw observed signal.

Shi and Weigend [1] explore discrete hidden states, and show their usefulness for characterizing and predicting very noisy time series. This is an extension of hidden Markov models, very popular in the speech community, but hardly known in the prediction community. The key idea is: if there are different dynamics in different regimes of the time series, and these regimes last for a while, then rather than averaging over the submodels, a more

appropriate model is obtained by estimating both the regime, and the parameters of the sub-models.
The MATLAB code we wrote for these experiments is available upon request.
The power of hidden Markov models crucially depend on the time series nature of the problem. Clearning, in contrast, as well as the "gated experts" architecture (Weigend, Mangeas, and Srivastava 1996) do not exploit the time series structure and are thus both more broadly applicable and weaker.

Timmer and Weigend [2] show the power of modeling dynamic noise and observational noise separately. I had mentioned previously (Section 2.1 of the progress report) that noisy inputs can lead to an underestimation of the parameters. This paper explores this point further and shows that a case where the decay times of shocks are underestimated by two orders of magnitude when the distinction between observational and dynamic noise is ignored. While state space modeling is a powerful method, it crucially depends on the time series nature of the problem.

Another method, suggested in the progress report, is blind source separation, related to independent component analysis (ICA). In collaboration with Dr. Andrew Back I started to explore the usefulness of independent component analysis (ICA, also called blind source separation) to very noisy data, Japanese stock return, in comparison to principal component analysis (PCA). Preliminary results indicate that estimated independent components (ICs, also called "sources") fall into two distinct categories: (1) a small number of large transient shocks (with skewed distributions), and (2) approximately Gaussian random noise.

Finally, the revision of a third paper by LeBaron and Weigend [3] focusing on focuses on performance evaluation by re-sampling, profited from the distinction of different noise sources: the method described in [2] was applied to that time series of daily NYSE volume.

In summary, while these papers received attention at several conferences and workshops, and have been accepted by major journals, the answer to the first stage of the clearning question has, unfortunately, been largely negative. I currently do not see a way to extend the algorithm to non-time-series data as I had hoped: there simply is not enough information for the degrees of freedom of both moving the data and the model.

3. Publications

[1] Shanming SHI and Andreas S. WEIGEND "Taking Time Seriously: Hidden Markov Experts Applied to Financial Engineering." In: Proceedings of the IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering (CIFEr, New York, March 1997), pp. 244--252. Piscataway, NJ: IEEE Service Center.

http://www.stern.nyu.edu/~aweigend/Research/Papers/HiddenMarkov/

Abstract--Most traditional time series models are global models based on local time information: they assume that the state can be fully and locally (in time) characterized with a finite embedding space. Prediction then amounts to simple regression. Unfortunately, there are many situations in which simple regression is not sufficient to model the temporal structure in a time series. We here introduce an architecture that we call Hidden Markov Experts. It is based on Hidden Markov Models used in speech recognition research. By introducing the concept of hidden states, Hidden Markov experts model time dependency of time series explicitly as a first-order Markov model with transitions between these hidden states. Within each state, local models are applied to estimate the probability density, which can be linear or nonlinear depending on the situation. This paper first discusses the statistical framework and the learning algorithm of Hidden Markov experts, then applies them to daily S&P500 data and to high frequency currency exchange rate data. The Hidden Markov Experts have better profit than the linear and nonlinear global models. The volatilities of the time series can be characterized by the hidden states.

[2] Jens TIMMER and Andreas S. WEIGEND "Exploiting Local Relations as Soft Constraints to Improve Forecasting." Forthcoming in: International Journal of Neural Systems, Vol. 8 (1997).

http://www.stern.nyu.edu/~aweigend/Research/Papers/StateSpace

Abstract--In time series problems, noise can be divided into two categories: dynamic noise which drives the process, and observational noise which is added in the measurement process, but does not influence future values of the system. In this framework, empirical volatilities (the squared relative returns of prices) exhibit a significant amount of observational noise. To model and predict their time evolution adequately, we estimate state space models that explicitly include observational noise. We obtain relaxation times for shocks in the logarithm of volatility ranging from three weeks (for foreign exchange) to three to five months (for stock indices). In most cases, a two-dimensional hidden state is required to yield residuals that are consistent with white noise. We compare these results with ordinary autoregressive models (without a hidden state) and find that autoregressive models underestimate the relaxation times by about two orders of magnitude due to their ignoring the distinction between observational and dynamic noise. This new interpretation of the dynamics of volatility in terms of relaxators in a state space model carries over to stochastic volatility models and to GARCH models, and is useful for several problems in finance, including risk management and the pricing of derivative securities.

[3] Blake LeBARON and Andreas S. WEIGEND "A Bootstrap Evaluation of the Effect of Data Splitting on Financial Time Series." Forthcoming in: IEEE Transactions on Neural Networks, Vol 9 (1998).

http://www.stern.nyu.edu/~aweigend/Research/Papers/Bootstrap/

Abstract: This article exposes problems of the commonly used technique of splitting the available data into training, validation, and test sets that are held fixed, warns about drawing too strong conclusions from such static splits, and shows potential pitfalls of ignoring variability across splits. Using a bootstrap or resampling method, we compare the uncertainty in the solution stemming from the data splitting with neural network specific uncertainties (parameter initialization, choice of number of hidden units, etc.). We present two results on data from the New York Stock Exchange. First, the variation due to different resamplings is significantly larger than the variation due to different network conditions. This result implies that it is important to not over-interpret a model (or an ensemble of models) estimated on one specific split of the data. Second, on each split, the neural network solution with early stopping is very close to a linear model; no significant nonlinearities are extracted.

4. Presentations

Time Series Analysis and Financial Modeling Johns Hopkins (Baltimore, Jan 9, 1998)

Modeling Volatility Using State Space Models (London, Dec 17, 1997)

Finding Hidden Structure in Financial Time Series NBER Summer Institute (Cambridge, MA, Jul 16, 1997)

Data Mining in Finance IBM Research (Yorktown Heights, Jun 11, 1997)

Learning from Data in Finance and Business Leonard N. Stern School of Business, Affiliates Seminar (Mar 27, 1997)

Time Series Tools Computational Intelligence in Financial Engineering (New York, Mar 22, 1997)

Hidden Markov Experts RIKEN (Tokyo, Nov 1, 1996)

New Architectures for Time Series Analysis Neural Networks for Signal Processing (Keynote Lecture) (IEEE-NNSP, Kyoto, Sep 4, 1996)

Taking Time Seriously: The State of the State Department of Mathematical Engineering and Information Physics, University of Tokyo (Aug 29, 1996)

Neural Networks in Financial Engineering Monash University, Department of Business Systems (Melbourne, Jul 15, 1996)

Time Series and Chaos International Mathematical Society Meeting (IMS, Sydney, Jul 11, 1996)

Nonparametric Statistics: The Road Ahead Australian National University, Statistics Department (ANU, Canberra, Jul 7, 1996)

5. Other (Interactions, transitions, patent disclosures, etc.)

Due to my leaving CU and relinquishing the remainder of the first year of the grant, there unfortunately were no interactions, transitions, patent disclosures, or honors.

6. Personnel Supported

Andreas Weigend (PI) 3 months
Shanming Shi (graduate student) 2 1/2 months
Mark Choey (graduate student) 1 month
Mike Fellows (computer support) 1 month
Pat Libhart (secretary, 5% for 1 year)

[illegible], NYC Dec 10, 97

Working Paper IS-97-002, Leonard N. Stern School of Business, New York University.
In: *Proceedings of the IEEE/IAFE Conference on Computational Intelligence for Financial Engineering (CIFEr'97, New York, March 1997),* pp. 244–252. Piscataway, NJ: IEEE Service Center.
http://www.stern.nyu.edu/ aweigend/Research/Papers/HiddenMarkov

Taking Time Seriously: Hidden Markov Experts Applied to Financial Engineering

Shanming Shi*
Department of Computer Science
University of Colorado
Campus Box 430
Boulder, CO 80303
shanming@cs.colorado.edu
http://www.cs.colorado.edu/~shanming

Andreas S. Weigend
Department of Information Systems
Leonard N. Stern School of Business
44 West Fourth St., MEC 9-74
New York University, New York, NY 10012
aweigend@stern.nyu.edu
http://www.stern.nyu.edu/~aweigend

Abstract. Most traditional time series models are global models based on local time information: they assume that the state can be fully and locally (in time) characterized with a finite embedding space. Prediction then amounts to simple regression. Unfortunately, there are many situations in which simple regression is not sufficient to model the temporal structure in a time series. We here introduce an architecture that we call *Hidden Markov Experts.* It is based on Hidden Markov Models used in speech recognition research. By introducing the concept of hidden states, Hidden Markov experts model time dependency of time series explicitly as a first-order Markov model with transitions between these hidden states. Within each state, local models are applied to estimate the probability density, which can be linear or nonlinear depending on the situation. This paper first discusses the statistical framework and the learning algorithm of Hidden Markov experts, then applies them to daily S&P500 data and to high frequency currency exchange rate data. The Hidden Markov Experts have better profit than the linear and nonlinear global models. The volatilities of the time series can be characterized by the hidden states.

Keywords. Regime Switching, Hidden States, Probability Density Prediction, Non-constant Transition Probabilities, EM Algorithm, Risk Estimation, Decision Technology.

Data sets used. High-frequency DEM/USD exchange rates. Daily S&P 500.

1 Introduction

Basic linear time series models are global models based on local time information and are typically based on two assumptions: (1) stationarity or weak stationarity of the time series, (2) the time series can be fully and locally (in time) characterized within a finite embedding space. However, many financial time series are certainly not stationary. In particular, they tend to have either time-varying means or variances or both, and for high frequency data, have varying dynamics during the day. Some of these problems are addressed for example through the family of autoregressive conditional heteroskedastic (ARCH) processes, assuming that the variance of the time series conditionally depends on past variances.

An important class of nonstationarity is piece-wise stationarity where the time series switches between different regions. Within the regions, the time series satisfy the requirement of stationarity, but between them, they might have different noise levels or different dynamics. Examples of such models are threshold autoregressive models and stochastic volatility models. Although a single global model can theoretically express any relationship including regime switching, it is often very hard to estimate such a global model from the data. Many architectures have been proposed to solve the problem of regime switching (e.g., [Jacobs et al., 1991, Weigend et al., 1995]), which decompose the global model into modular local models for the regions. However, the key point is how to split the data space. In these models, however, the regions are assumed to be independent of each other, i.e., if we shuffle the patterns of the data set, there will be no difference in the final model.[1]

In this paper, we use Hidden Markov Experts to explicitly model the time dependency between adjacent

*The author is currently with J.P.Morgan & Co. Inc., 60 Wall St, New York, NY, 10260
shi_shanming@jpmorgan.com

[1] In this paper, the word *pattern* denotes an input-output pair.

patterns of the time series. HMMs have been widely used in the field of speech recognition where context is important [Rabiner, 1989]. It can also be used in modeling the time dependency of regime switching. Related work in this field is Hamilton's regime switching model [Hamilton, 1990]. However, in Hamilton's work the regions or the states can be directly estimated from the current observation, while in Hidden Markov Experts, the states are hidden from the observation and depend on the whole history of observations. There are several variations of Hamilton's work on regime switching, see Chapter 22 in [Hamilton, 1994]. They however all use linear predictors, whereas we allow for nonlinear predictors.

This paper is organized as follows: Section 2 explains the basic idea and formalism of Hidden Markov Experts. Section 3 reports the results of the experiments we carried out on computer-generated data (where we know the true segmentation), as well as on two real-world problems (high-frequency exchange rates and S&P 500). Section 4 describes the extension of non-constant transition probabilities, and Section 5 summarizes the results obtained.

2 Hidden Markov Experts

Any maximum likelihood approach needs several assumptions. First, a *noise model* has to be chosen; it describes how likely an observed data point is, given the model's prediction. The typical choice of minimizing the sum of squared errors corresponds to assuming Gaussian distribution for the noise model. Second, a choice about the *architecture, model class* or *functional form* between inputs and outputs has to be made. Typical examples are linear models for regression, logit for classification, or general nonlinear functions such as implemented by neural networks. Their output typically are expected values (possibly with variances) as predictions. In standard regression or classification cases, any dependencies between patterns are ignored. The third assumption now addresses precisely these dependencies between patterns: we here use a Hidden Markov model to model the relation between adjacent patterns.

2.1 Hidden Markov Models

Basic Idea: The observed sequence of observations is determined by the underlying unobservable stochastic process, the state sequence of the HMM, with an *emission probability.* A Hidden Markov model is called *hidden* because these states can not be directly estimated from the observed data. We also assume that the hidden process is a *Markov* process: the probability of the next state depends only on the current state and the transition probability between the two sates. Both the states and the observed process can be either discrete or continuous. For time series modeling, we use discrete states (corresponding to the regimes) and continuous observations (corresponding to the time series).

Notation: (1) Observations (time series data): $\mathcal{Y} = \{y^1, y^2, ..., y^T\}$, (2) States: $\mathcal{S} = \{s_1, s_2, ..., s_M\}$, (3) Transition probabilities: $\mathcal{A} = \{a_{ij}, i, j \in M, a_{ij} = Prob(\text{next state} = j | \text{current state} = i)\}$, (4) Emission probabilities: $\mathcal{B} = \{b_j^t, j \in M, t \in T, b_j^t = Prob(\text{current observation} = y^t | \text{current state} = j)\}$, (5) Initial probabilities of each state: $\Pi = \{\pi_i, i \in M\}$. For convenience, the notation $\lambda = \{\mathcal{A}, \mathcal{B}, \Pi\}$ is introduced to denote the entire set of parameters.

The Maximum Likelihood Function: The central problem of HMMs is to find the parameters λ that most likely fit the observed data $\mathcal{Y}$. Under our assumption of the time dependency between patterns, the probability $P(y^t|\lambda)$ is *not independent* of time t, for pattern at time t. However the joint probability $P(y^t, s^t|\lambda)$ is *independent* of time t. The likelihood $P(\mathcal{Y}|\lambda)$ is then given as

$$P(\mathcal{Y}|\lambda) = \sum_{\forall Q} P(\mathcal{Y}, Q|\lambda) = \sum_{\forall Q} \prod_{t=1}^{T} P(y^t, s^t|\lambda) = \sum_{\forall Q} \prod_{t=1}^{T} P(y^t|s^t, \lambda) P(s^t|\lambda) \tag{1}$$

where Q is the state sequence corresponding to each pattern, e.g. $Q_A = \{s_{t1}, s_{t2}, ...s_{ti}, ..., s_{tT} | s_{ti} \in S\}$ and $P(y^t|s_j^t, \lambda) = b_j^t$. Therefore, to get the probability $P(\mathcal{Y}|\lambda)$, two probabilities need to be estimated: One is the probability of current state, the other is the emission probability given the current state. Since there are M^T combinations of different Q's, it is very difficult to take the derivative of equation (1) with respect to λ. An algorithm to do this is called the *forward-backward* procedure can be used to efficiently calculate the $P(\mathcal{Y}|\lambda)$ [Baum, 1972, Rabiner, 1989].

2.2 Experts: models for emission probabilities

Now we can specify the architecture for emission probabilities. If we assume the input of the emission model is $\mathcal{X} = \{x^t,\ t \in 1...T\}$, then the emission probability $b_j^t = P(y^t|x^t, s_j^t, \lambda)$. This is the likelihood of observing data given the current state and the current input. We call each of the specified emission models an expert and each individual expert corresponds to one state.

The experts can take on different architectures. For instance, we can use a linear model or a nonlinear model, such as a neural network. Furthermore, different experts can have different sets of inputs. This turns out to be an important advantage that alleviates the effects of the *curse of dimensionality*. In this paper, we are going to use the neural and the AR model as the experts. Instead of emission probability $\mathcal{B}$, we have a new set of parameters $\Theta_{\mathcal{B}}$ for the emission model. The emission probability $\mathcal{B}$ can be computed from $\Theta_{\mathcal{B}}$. One can then compute $\lambda = \{\mathcal{A},\ \Theta_{\mathcal{B}},\ \Pi\}$.

2.3 Learning algorithm

Baum and his colleagues [Baum, 1972] proposed an elegant algorithm called the *forward-backward* procedure to calculate $P(\mathcal{Y}|\lambda)$. They also introduced an EM (Expectation Maximization) algorithm to maximize this probability. We here generalize these algorithms to Hidden Markov Experts.

- **Forward-backward procedure**: Define $\alpha_i^t = P(y^1, y^2, ..., y^t, s_i^t|\lambda)$, where $1 \leq t \leq T$. Then we obtain for the probability $P(\mathcal{Y}|\lambda) = \sum_{i=1}^M \alpha_i^T$. The α_i^T can be calculated through the recursive procedure: $\alpha_i^1 = \pi_i b_i^1$ and $\alpha_i^{t+1} = [\sum_{i=1}^M \alpha_i^t a_{ij}] b_j^{t+1}$. This is called the forward procedure. Similarly, we can define the backward variable $\beta_i^t = P(y^{t+1}, y^{t+2}, ...y^T|s_i^t, \lambda)$. With the recursive induction $\beta_i^T = 1$ and $\beta_i^t = \sum_{j=1}^M a_{ij} b_j^{t+1} \beta_j^{t+1}$, where $t = T-1, T-2, ..., 2, 1$, we can get all the β for each t. The reason we need this backward procedure is to use the whole observed sequence to estimate the probability $P(s_j^t|\lambda)$. With α and β, we can determine the γ_i^t defined as $P(s_i^t|\mathcal{Y}, \lambda)$

$$\gamma_i^t = \frac{P(\mathcal{Y}, s_i^t|\lambda)}{P(\mathcal{Y}|\lambda)} = \frac{\alpha_i^t \beta_i^t}{P(\mathcal{Y}|\lambda)} = \frac{\alpha_i^t \beta_i^t}{\sum_{k=1}^M \alpha_k^t \beta_k^t} \tag{2}$$

  The probability γ_i^t can be used as the estimation of $P(s_i^t|\lambda)$, since it is the best we can do given the whole observation sequence. Similarly an auxiliary probability, $\xi_{ij}^t = P(s_i^t, s_j^{t+1}|\mathcal{Y}, \lambda)$, can also be computed with α and β as

$$\xi_{ij}^t = \frac{P(s_i^t, s_j^{t+1}, \mathcal{Y}|\lambda)}{P(\mathcal{Y}|\lambda)} = \frac{\alpha_i^t a_{ij} b_j^{t+1} \beta_j^{t+1}}{\sum_{i=1}^M \sum_{j=1}^M \alpha_i^t a_{ij} b_j^{t+1} \beta_j^{t+1}} \tag{3}$$

- **EM algorithm:** In the **expectation step**, the probability α and β, and in turn, the posterior γ and ξ for each t, are calculated based on the current estimation of λ according to (2) and (3). In the **maximization step**, we update the $\lambda = \{\mathcal{A}, \Theta_{\mathcal{B}}, \Pi\}$ according to $\pi_i = \gamma_i^1$ and

$$a_{ij} = \frac{\text{expected number of transitions from state i to j}}{\text{expected number of transitions from state i (to anywhere)}} = \frac{\sum_t \xi_{ij}^t}{\sum_t \gamma_i^t}$$

  For each emission model, maximizing equation (1) is the same as maximizing the following:

$$P(\mathcal{Y}, s^t = j, \forall\ t|\mathcal{X}, \lambda) = \prod_{t=1}^T P(y^t|x^t, s_j^t, \Theta_{\mathcal{B}}^j) \gamma_j^t \tag{4}$$

  where the $\Theta_{\mathcal{B}}^j$ represents the parameters of the emission model of state j. Equation (4) or its negative logarithm can be viewed as a cost function for the emission model. The way to compute the parameter $\Theta_{\mathcal{B}}^j$ depends on the format of the experts and the assumption of the noise. For example, in our experiment we assume the error to be Gaussian distributed and use neural networks as experts. Equation (4) is then similar to weighted sum-squared error and the parameter $\Theta_{\mathcal{B}}^j$ includes the weights and bias of the network, as well as the variance of the data in state j.

2.4 Making prediction

For financial engineering, the key question is how to make predictions with Hidden Markov Experts. In prediction, we cannot use Equation (2) to estimate the state, because it includes future information. However, given the sequence of observations up to now, we can estimate the probability of state in terms of the transition probabilities a_{ij} and α as

$$P(s_j^{t+1}|y^1, y^2, ..., y^t, \lambda) = \frac{P(y^1, y^2, ..., y^t, s_j^{t+1}|\lambda)}{P(y^1, y^2, ..., y^t|\lambda)} = \frac{\sum_{i=1}^{M} \alpha_i^t a_{ij}}{\sum_{j=1}^{M}(\sum_{i=1}^{M} \alpha_i^t a_{ij})} \quad . \tag{5}$$

The expected value of the prediction then becomes $\hat{y}^{t+1} = \sum_{j=1}^{M} \hat{y_j}^{t+1} P(s_j^{t+1}|y^1, y^2, ..., y^t, \lambda)$.

3 Experiments

We present one computer simulated data set and two real world data sets, which are the high frequency Olsen foreign exchange data and the S&P500 daily data. For the real world data, we compute the profit based on the sign of the prediction, i.e., buy if the sign of the prediction of the next return is positive and sell if the sign of the prediction of the next return is negative. Each data set is split into a training set and a test set. All the results given are obtained on the test set.

3.1 Computer simulated data

To convince ourselves of the applicability of the idea, we generate a time series that switches between a trending and a mean reverting process with i.i.d. Gaussian innovations. The diagonal transition probabilities of the transition matrix are $a_{11} = 0.98$, and $a_{22} = 0.97$. The trending process is $r^{t+1} = 0.2r^t + 0.8\,\mathcal{N}(0,1)$, and the mean reverting process is $r^{t+1} = -0.15r^t + 0.5\,\mathcal{N}(0,1)$. These two are high noise processes where the signal-to-noise ratio is 0.04 and 0.0225 respectively.

Two AR experts are used in this experiment. In contrast to real world data, we know the true segmentation of the data set. The experiment shows that the hidden Markov experts recovered the regimes and the parameters including the variances of the two processes. Figure 1 shows the segmentation of one expert on out-of-sample data compared to the true segmentation. Table 1 gives the statistics of the parameter estimations: the transition probabilities A, the AR coefficients κ, and the standard deviations of Gaussian noise σ over 20 different runs.

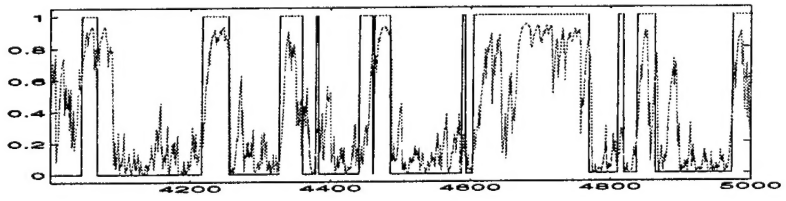

Figure 1: Regimes found by one expert compared to the true segmentation on out-of-sample simulated data. The solid line shows the true value used when the data was generated, the dotted line the causal prediction of the model.

	a_{11}	a_{22}	κ_1	κ_2	σ_1	σ_2
true value	0.980	0.970	0.200	−0.150	0.800	0.500
mean of fitted	0.981	0.971	0.193	−0.140	0.804	0.498
std of fitted	0.002	0.004	0.015	0.026	0.006	0.008

Table 1: Summary of the experiments on the computer simulations. For the transition probabilities on the main diagonal a_{ii}, the AR coefficients κ_i, and the noise levels σ, we give the true values as well as the mean and standard deviation of their estimates through 20 runs of the model.

3.2 High frequency foreign exchange data

The first real world data set is part of Olsen's DEM/USD foreign exchange rate data based on a variable time scale, called ϑ time, instead of fixed intervals of physical time [Dacorogna et al., 1996]. We model these data with three experts all using five lagged values of the time series as the input to predict the next value. We compare the results with a global linear model (the AR model) and a global nonlinear model (a feed-forward neural network). The neural network has one linear output and 10 tanh hidden units. The training set contains 1000 points from 19/05/95 17:58 to 09/06/95 14:32. The test set contains 1000 points from 09/06/95 14:41 to 29/06/95 23:54. The Hidden Markov Experts are trained for one-step ahead predictions (i.e., half an hour in ϑ time).

Figure 2 shows the results on the test set. The top panel gives the data, the central panel the absolute values of the price returns, and the bottom panel shows the segmentation of the three experts for the Olsen test data. Comparing the lower two panels note that the first expert tends to take the regimes with relatively low volatility, the second expert tends to take the regimes with relatively high volatility, and the third expert takes care of the outliers.

When we estimate the parameters of Hidden Markov Experts, we also estimate the variance of each expert. They are plotted as a function of training time in Figure 4. Note that the variance of the first expert is only about a quarter of the variance of the second expert. This can be associated with low and high-volatility regimes. The third expert rarely gets activated in response to large returns.

Figure 3 shows the profit and loss curves of the Hidden Markov Experts after 50 training iterations, in comparison to a simple feed-forward neural network, linear regression, and to a simple "short-and-hold" method. The Hidden Markov Experts have the highest profit over the period of the test set.

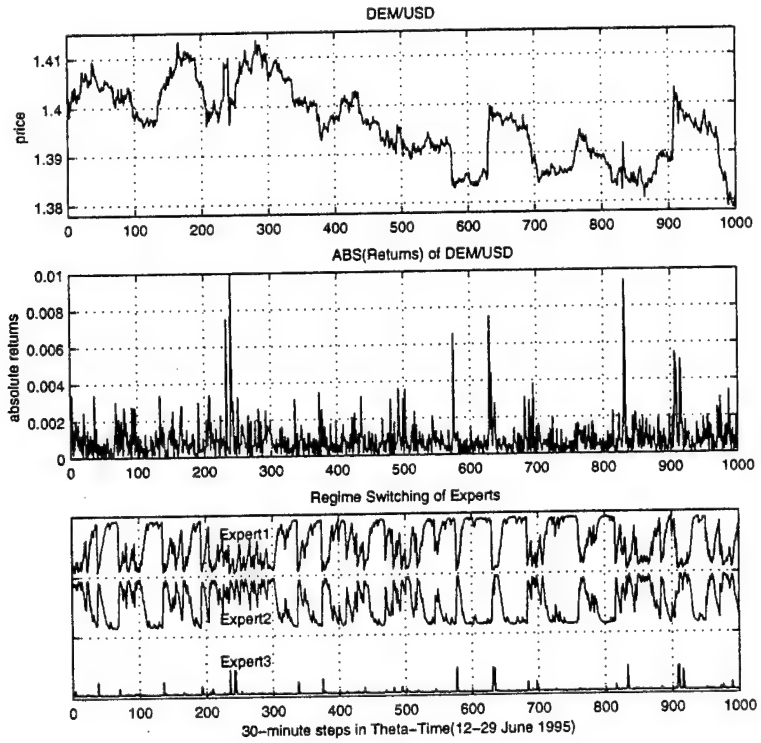

Figure 2: Panel 1 of this figure is the test set of the Olsen data. Panel 2 shows the absolute values of the return of the price. Panel 3 shows the regimes found by each experts for the Olsen data. We plot the responses of all three experts in this panel with different offsets. We can see that the first expert is responsible for the low volatility regions, while the second expert is responsible for high volatility regions, and the third expert acts as a collector for the outliers. This result is consistent with the variances of each expert, given in Figure 4.

An important feature of this architecture is that it gives the entire *probability density* of the return–not just

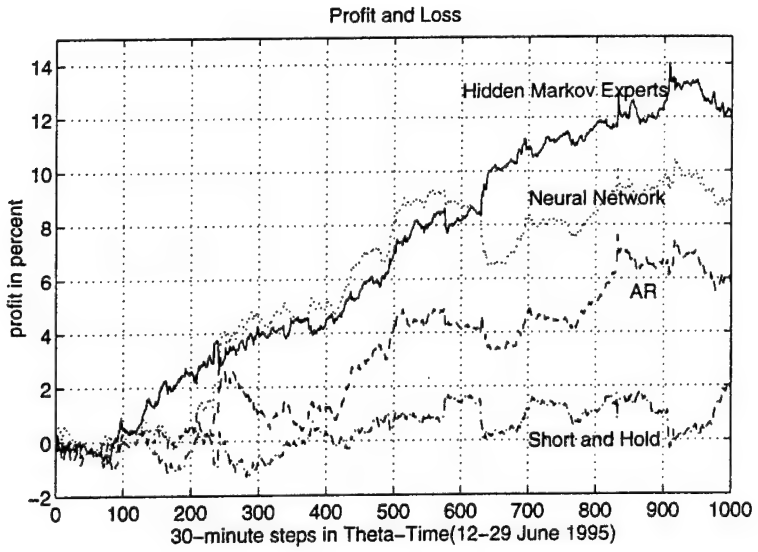

Figure 3: Profit and Loss for Olsen data. This figure shows that the Hidden Markov Experts give the highest profit in comparison to the pure linear regression and the feed-forward neural network. Transaction cost are not taken into account.

a prediction of its expected value. An example is given in Figure 5, where the solid line is the mixture and the dashed lines are the individual Gaussians. The full density can subsequently be used in the computation of risk [Weigend et al., 1997].

3.3 Daily S&P500

The second real world experiment compares the results of the linear Hidden Markov Experts with nonlinear Hidden Markov Experts (using neural networks as experts). Two experts have been used for both the linear and nonlinear Hidden Markov Experts. Each neural network expert has one linear output unit and 10 tanh hidden units. The training set spans from 01/12/73 to 12/31/86 and the test set spans from 01/02/87 to 12/29/94. Figure 6 shows the S&P 500 data, the daily returns, and the segmentation found by nonlinear Hidden Markov Experts. The plotted regime corresponds to the low volatility regions, its complement to high volatility regions. The Figure 7 shows the profit and loss curves of the different methods. The nonlinear Hidden Markov Experts have better profit than the linear Hidden Markov Experts and the AR model in the test period.

4 Time-varying Transition Probabilities

We extend the work described so far by relaxing the assumption of the a_{ij}'s being constant: we assume that the transition probabilities vary depending upon some external inputs. This is a reasonable assumption for modeling the complex financial market and the influence of different kind of economic indicators.

Our extension can be compared to Bengio and Frasconi's "Input Output Hidden Markov Model," using a recurrent mixture of experts to estimate the transition probabilities [Bengio and Frasconi, 1996]. The recurrent architecture and many parameters lead to problems in convergence and overfitting. The problem is that we have no target for the transition probabilities. Here we provide a new and simpler way to implement the idea of nonstationarity based on Hidden Markov Experts.

Instead of directly estimating the transition probability $P(s_i^t|s_j^t)$, we can model the time dependency of the joint probability $P(s_i^t, s_j^t)$. According to Equation (3), we can compute the posterior probability $\xi_{ij}^t = P(s_i^t, s_j^{t-1}|\mathcal{Y}, \lambda)$ with α and β for each time t. Then with Bayes rule, we obtain the transition probability

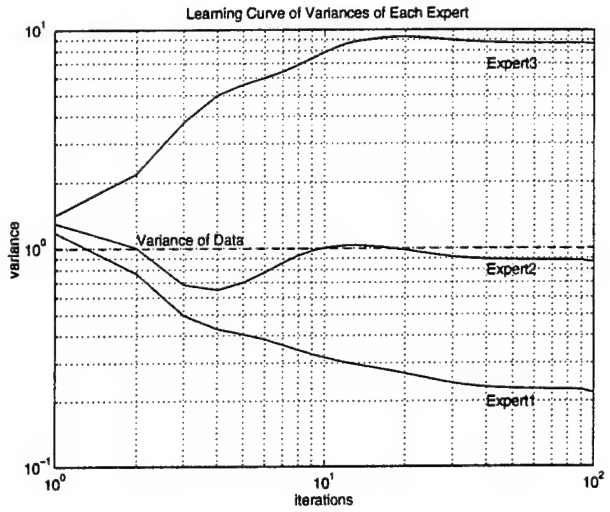

Figure 4: Learning curve of the variance for each expert for the Olsen data. We can see that the first expert is responsible for the low volatility regions, while the second expert is responsible for high volatility regions (similar to the variance of the time series). The third expert acts as a collector for the outliers. (The data is normalized to unit variance.)

at time t

$$\mathcal{A}_{ij}^t = \frac{P(s_i^t, s_j^{t+1}|\mathcal{Y}, \lambda)}{\sum_{i=1}^{M} P(s_i^t, s_j^{t+1}|\mathcal{Y}, \lambda)} \tag{6}$$

In order to estimate the transition probability with extra inputs, the architecture can be either a linear model or a nonlinear model. The target of this local model is ξ_{ij}^t at time t. Since ξ_{ij}^t is a probability, we can use softmax to meet the constraint. We have applied this model to computer-generated data with known time-varying transition probabilities. The architecture and algorithm correctly estimates the time dependency of the transition probabilities.

5 Conclusions

This paper introduced the theory and architecture of Hidden Markov Experts. We presented several experiments on financial time series. The key results are:

1. We can find clean segmentation into a small number of experts. There is no way of determining an "optimal" number of experts from first principles and the data. For real world problems, this is one of the degrees of freedom in modeling.
2. We show that the segmentation can be interpreted in terms of volatility. We carried out tests randomly flipping the sign of the returns, yielding similar segmentation. However, for these randomized returns the profit disappears as expected.
3. We compare the Hidden Markov Experts to linear models and to a simple buy-and-hold strategy. On all data sets we tested, both profit and Sharpe Ratio are better for the Hidden Markov Experts than for the benchmarks.
4. We also compare Hidden Markov Experts with nonlinear emissions models to those with linear emission models. When properly controlled for overfitting, the nonlinear emission models outperform the linear ones.
5. We extend the standard framework of constant transition probabilities to conditional transition probabilities.
6. An important application for this architecture is risk management. The algorithm gives the probability density of the return–not just a prediction of its expected value! The density is expressed as a mixture of Gaussians and can be used in the computation of various risk measures.

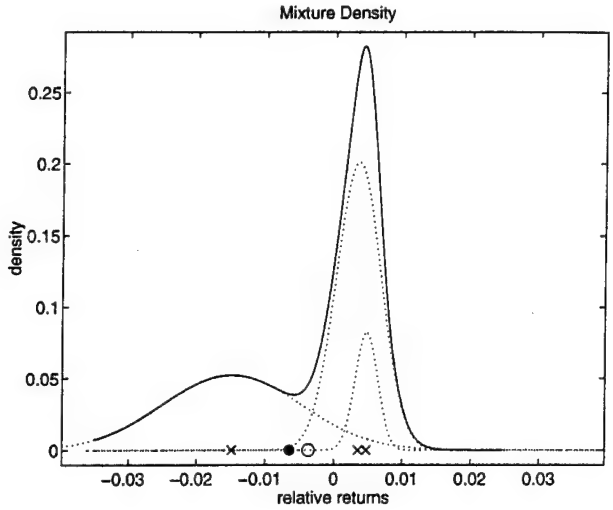

Figure 5: Predicted probability density function of the returns for a specific half-hour prediction of the test set of the Olsen data. The solid curve in this figure shows the mixture of the three Gaussian densities. The individual densities of each state are shown as dashed curves. The circle corresponds to the mean prediction, and the solid dot is the target. The individual mean of each expert is shown by the x's. (We did not normalize the figure to integrate to unity; the curves are just proportional to the density.)

Acknowledgments

We would like to thank Olsen and Associates for allowing us to use their foreign exchange data, as well as Hal White for kindly providing us with the S&P500 data. Shanming Shi is grateful for the hospitality and discussions at the Information Systems Department at NYU's Leonard N. Stern School of Business. Andreas Weigend acknowledges support from the National Science Foundation (ECS-9309786) and the Air Force Office of Scientific Research (F49620-96-1-0240). Please contact the authors for the availability of the MATLAB programs.

References

[Baum, 1972] Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov process. *Inequalities*, 3:1–8.

[Bengio and Frasconi, 1996] Bengio, Y. and Frasconi, P. (1996). Input-output HMM's for sequence processing. *IEEE Transactions on Neural Networks*, 7:1231–1249.

[Dacorogna et al., 1996] Dacorogna, M. M., Gauvreau, C. L., Müller, U. A., Olsen, R. B., and Pictet, O. V. (1996). Changing time scale for short-term forecasting in financial markets. *Journal of Forecasting*, 15:203–227.

[Hamilton, 1990] Hamilton, J. D. (1990). Analysis of time series subject to changes in regime. *Journal of Econometrics*, 45:39–70.

[Hamilton, 1994] Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press, Princeton.

[Jacobs et al., 1991] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3:79–87.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286.

[Weigend et al., 1997] Weigend, A. S., Abu-Mostafa, Y. S., and Refenes, A.-P. N., editors (1997). *Decision Technologies for Financial Engineering (Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets, NNCM-96)*. World Scientific, Singapore.

[Weigend et al., 1995] Weigend, A. S., Mangeas, M., and Srivastava, A. N. (1995). Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. *International Journal of Neural Systems*, 6:373–399.

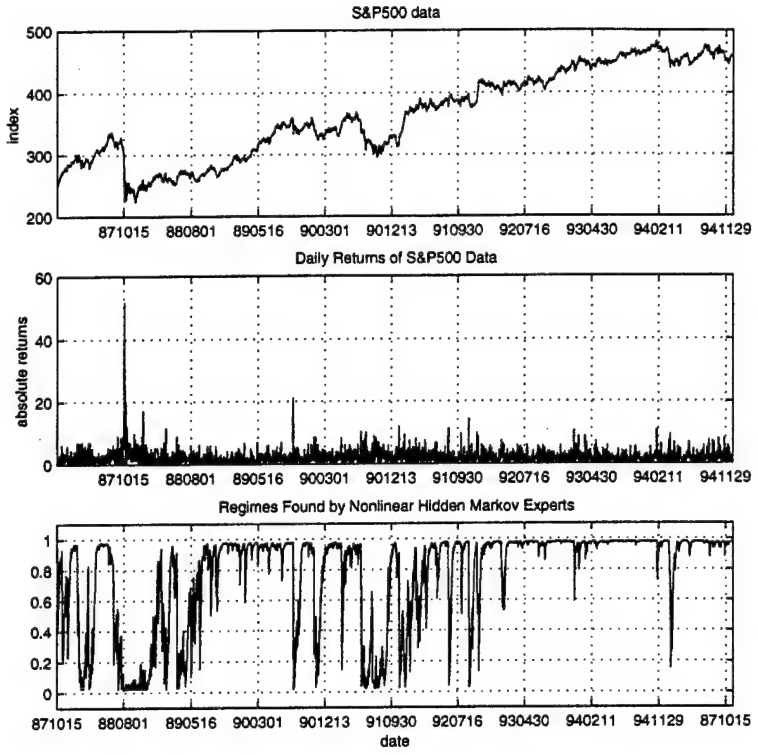

Figure 6: The first panel is the S&P500 data, the second panel shows the returns, the third panel is the state of the low volatility regions.

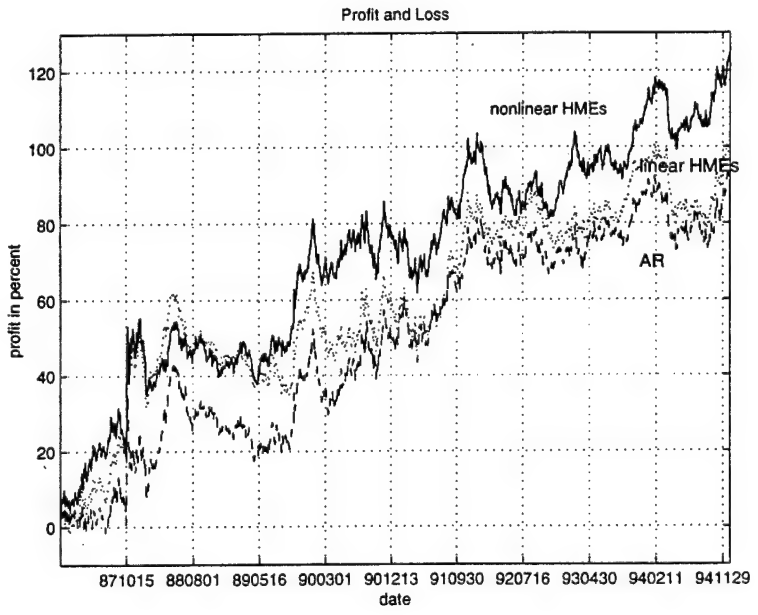

Figure 7: This figure shows the profit and loss of the S&P500 data. The nonlinear Hidden Markov Experts (nonlinear HMEs) have better profit than the linear Hidden Markov Experts (linear HMEs) and the AR model.

[2]

Working Paper IS-97-11, Leonard N. Stern School of Business, New York University.
Forthcoming in: *International Journal of Neural Systems*, Vol. 8 (1997).
(Special Issue on "Modeling Noisy Time Series.")
http://www.stern.nyu.edu/~aweigend/Research/Papers/StateSpace

MODELING VOLATILITY USING STATE SPACE MODELS

Jens Timmer
Fakultät für Physik
Universität Freiburg
Hermann-Herder Strasse 3
D–79104 Freiburg, Germany
jeti@fdm.uni-freiburg.de
www.fdm.uni-freiburg.de/~jeti

Andreas S. Weigend
Department of Information Systems
Leonard N. Stern School of Business
New York University
44 West Fourth Street, MEC 9-74
New York, NY 10012, USA
aweigend@stern.nyu.edu
www.stern.nyu.edu/~aweigend

Abstract. In time series problems, noise can be divided into two categories: dynamic noise which drives the process, and observational noise which is added in the measurement process, but does not influence future values of the system. In this framework, empirical volatilities (the squared relative returns of prices) exhibit a significant amount of observational noise. To model and predict their time evolution adequately, we estimate state space models that explicitly include observational noise. We obtain relaxation times for shocks in the logarithm of volatility ranging from three weeks (for foreign exchange) to three to five months (for stock indices). In most cases, a two-dimensional hidden state is required to yield residuals that are consistent with white noise. We compare these results with ordinary autoregressive models (without a hidden state) and find that autoregressive models underestimate the relaxation times by about two orders of magnitude due to their ignoring the distinction between observational and dynamic noise. This new interpretation of the dynamics of volatility in terms of relaxators in a state space model carries over to stochastic volatility models and to GARCH models, and is useful for several problems in finance, including risk management and the pricing of derivative securities.

Data sets used. Olsen & Associates high frequency DEM/USD foreign exchange rates (8 years). Nikkei 225 index (40 years). Dow Jones Industrial Average (25 years).

1 Introduction

Modeling and predicting the volatility of financial time series has become one of the central areas in finance and trading; examples range from pricing derivative securities to computing the risk of a portfolio. Volatility is usually predicted using generalized autoregressive conditional heteroskedastic (GARCH) models; Bollerslev, Engle and Nelson (1995) guide through the GARCH literature, and Engle (1995) collects some of the key papers.

Here we present an alternative to GARCH that models the underlying dynamics using a state space model. This allows us to describe the hidden process in terms of variables natural for a dynamic system, such as decay times for shocks, its spectrum, and the dimensionality of the underlying process. Stochastic volatility models (see Shephard (1996) for a review) are a variant of the general state space approach presented here. They differ in that the mapping from the hidden variable to the observed variable is nonlinear. The interpretation developed in this article can also be helpful for understanding and characterizing stochastic volatility models.

This article is organized as follows: Section 2 discusses observational noise and dynamic noise, and reviews intuitions and interpretations for linear systems, important for understanding the results in physical terms, such as decay times of volatility shocks. Section 3 defines and explains the formalism of state space models. Variations and interpretations that are typical in finance and in econometrics are given in Section 4. Section 5 describes the three data sets used for the empirical studies. The results are presented in Section 6, and the effect of ignoring existing observational noise on the model is discussed in Section 7. Section 8 summarizes the findings and discusses some of the applications of this approach for noisy time series in finance.

2 Some background concepts

2.1 Observational noise and dynamic noise

In time series modeling, one crucial question is whether or not observational noise is present in the data. Observational noise of a high level can pose a severe problem if it is not treated properly, leading to models that underestimate the functional relation between past and future values. A typical example of such *observational noise* is when an astronomer observes a star: fluctuations in the atmosphere, or a subway train passing by and shaking a telescope that points to the star, will not influence the dynamics of the star. In contrast, a noise component that does influence the dynamics of a system is called *dynamic noise*. For example, in an autoregressive process, the noise truly moves the state (sometimes also expressed as "the noise drives the system"), and subsequent values are derived from that moved state.

This article focuses on discrete time dynamics, typically modeled by difference equations or maps. The distinction between observational noise and dynamic noise is also important for continuous time dynamics, typically modeled by differential equations.

2.2 Interpretations of linear systems

To facilitate the interpretation of state space models (introduced in Section 3), we first review autoregressive processes without observational noise, and characterize them from several perspectives. A simple way of generating a time series is through an autoregressive (AR) process of order p, AR[p] (Yule 1927, Priestley 1981, Oppenheim and Schafer 1989)

$$x(t) = \sum_{i=1}^{p} a_i\, x(t-i) + \epsilon(t) \quad , \tag{1}$$

where $\epsilon(t)$ denotes an uncorrelated Gaussian distributed random variable with mean zero and constant variance σ^2, $\mathcal{N}(0, \sigma^2)$. Through the eyes of a physicist, such a process can be interpreted as a combination of *relaxators* and *damped oscillators* (Honerkamp 1993). The simplest case is an AR[1] process

$$x(t) = ax(t-1) + \epsilon(t) \quad . \tag{2}$$

It can be characterized in the time domain as a relaxator by an exponentially decaying impulse response, proportional to $\exp(-t/\tau)$, with the relaxation time

$$\tau = -\frac{1}{\log a} \quad . \tag{3}$$

After this time, the amplitude of an impulse will have decayed to 1/e or 37% of its initial value.

In the frequency domain, an AR process can be interpreted as a filter responding to white noise. The power spectrum of an AR[1] process drops off with

$$S(\omega) = \frac{\sigma^2}{|1 - ae^{-i\omega}|^2} = \frac{\sigma^2}{1 + a^2 - 2\cos\omega} \quad . \tag{4}$$

For an AR[2] process, there are two qualitatively different cases, depending on the values of the parameters. We can always rewrite a single AR[2] model as a set of two AR[1] models using the transformation

$$\mathbf{A} = \begin{pmatrix} a_1 & a_2 \\ 1 & 0 \end{pmatrix} \quad . \tag{5}$$

Its eigenvalues

$$\lambda_i = \frac{a_1}{2} \pm \sqrt{\frac{a_1^2}{4} + a_2} \tag{6}$$

characterize the behavior of the AR[2] process. If the eigenvalues are real ($a_1^2/4 + a_2 \geqslant 0$), the AR[2] process can be characterized as the superposition of two relaxators, and the spectrum drops off monotonically with increasing frequencies, again with decay constants

$$\tau_i = -\frac{1}{\log \lambda_i} \quad (i = 1, 2) \quad . \tag{7}$$

If the eigenvalues are complex, the AR[2] process describes a resonance, corresponding to a hump in the spectrum.[1] In both cases, the spectrum is given by

$$S(\omega) = \frac{\sigma^2}{|1 - a_1 e^{-i\omega} - a_2 e^{-2i\omega}|^2} \quad . \tag{8}$$

[1] For a damped oscillator (the case of complex eigenvalues), the parameters can be expressed through the characteristic period T and the relaxation time τ as

$$\begin{aligned} a_1 &= 2\cos\left(\frac{2\pi}{T}\right)\exp(-1/\tau) \\ a_2 &= -\exp(-2/\tau) \quad . \end{aligned}$$

By increasing the model order, an AR[3] process can combine a relaxator with an oscillator, and an AR[4] process can describe two oscillators, etc.

Despite the simplicity and multiple interpretability of AR models, not all processes in the world are linear autoregressive. Examples of generalizations without hidden states consist of including past q driving noise terms in the dynamics, yielding an autoregressive moving average ARMA$[p, q]$ processes,[2] as well as including nonlinearities.[3] Here we extend autoregressive models in a different direction, by allowing for a hidden state.[4] The next section introduces the notation and gives the formalism of state space modeling.

3 Formalism of linear state space models (LSSM)

In Eq. (1) the $x(t)$ served two roles: it was the variable that was observed, and it was the variable in which the dynamics was expressed. However, there are processes where the dynamics cannot be observed directly because it is masked by observational noise. Thus, no direct map exists from the observed data to the state. This requires the notion of a *hidden state*. In terms of notation, we keep the letter x as the variable that contains the dynamics, and use $y(t)$ for the observed variable. The state, characterized by the vector $\vec{x}(t)$, captures all the information needed to characterize the system at time t.

The key to state space modeling is to split the noise into two parts:

- dynamic noise $\vec{\epsilon}(t)$ that drives the evolution of the hidden state, and
- observational noise $\eta(t)$ that is a non-explainable additive contribution to the measured $y(t)$.

These contributions have been discussed in intuitive terms in Section 2.1. Their formal role can be seen by observing how they enter the two equations that describe a linear state space model (LSSM):

$$\vec{x}(t) = \mathbf{A}\,\vec{x}(t-1) + \vec{\epsilon}(t), \qquad \vec{\epsilon}(t) \in \mathcal{N}(0, \mathbf{Q}) \tag{9}$$

$$y(t) = \mathbf{C}\,\vec{x}(t) + \eta(t), \qquad \eta(t) \in \mathcal{N}(0, R) \quad . \tag{10}$$

Eq. (9) describes the dynamics. Eq. (10) maps the dynamics to the observation and includes the observational noise $\eta(t)$.

As in the case of the observable linear autoregressive model, discussed in Section 2.2, describing the process via physical quantities can yield important insights. The spectrum of a LSSM is given

[2]While for theoretical reasons ARMA$[p, p-1]$ should be preferred to AR$[p]$ processes for modeling of sampled continuous-time processes (Phadke and Wu 1974), we find that in practice, differences in the results are small.

[3]The linear mapping given by Eq. (1) can be generalized to become a nonlinear mapping. Note that this is fully within the autoregressive framework and amounts to simple regression. Nonlinear approaches include radial basis functions (Casdagli 1989, Moody and Darken 1989, Poggio and Girosi 1990), neural networks (Lapedes and Farber 1987, Weigend, Huberman and Rumelhart 1990), and nonparametric kernel methods (Tjostheim and Auestad 1994).

[4]This article explores the idea of a *continuous* hidden state, characterized by a scalar $x(t)$ or a vector $\vec{x}(t)$. The dynamics is expressed in terms of that unobserved state, and the state is subsequently mapped to the (conditional expectation of the) observed quantity. In contrast, Hidden Markov models (Rabiner 1989, Fraser and Dimitriadis 1994, Hamilton 1994, Bengio and Frasconi 1995, Shi and Weigend 1997) assume the hidden state to be *discrete*: for each of these hidden states, there is an "agent" or "expert" (e.g., expressed as an autoregressive model) that generates the next data point. This introduces a second level of dynamics that is described by the transitions between the hidden states. This level of dynamics is absent in a pure autoregressive framework.

by

$$S(\omega) = \mathbf{C}(1 - \mathbf{A}e^{-\mathrm{i}\omega})^{-1}\mathbf{Q}\left((1 - \mathbf{A}e^{\mathrm{i}\omega})^{-1}\right)^T \mathbf{C}^T + R \quad . \tag{11}$$

The superscript $(\cdot)^T$ denotes transposition. The spectra of AR processes, Eq. (8), are a subset of Eq. (11). Note that LSSM spectra include shapes that cannot be generated by AR processes. An important example of such a shape is a spectrum where for low frequencies the power drops similarly to an AR[1] process (see Eq. (4)), but for higher frequencies the power remains constant and does not continue to fall, as an AR model would require it to. This can be interpreted as a low-frequency process whose spectral energy decreases as the frequency increases, until it is masked by a noise floor of a noise source with a flat spectrum. This low-frequency signal above a flat noise floor is the crucial *spectral signature* of a LSSM that cannot be emulated by an ordinary autoregressive model.

While parameter estimation in AR models is well established (e.g., by the Burg or the Durbin-Levinson algorithms), it is more cumbersome in the case of state space models . A standard approach uses the expectation maximization (EM) algorithm (Dempster, Laird and Rubin 1977), a general iterative procedure for estimating parameters for models with hidden variables. In the E-step, it is assumed that the parameters of the model are known, and the hidden variables are estimated. In the M-step, the estimates of the hidden variables are taken literally and the values of the parameters are adjusted. This approach was first applied to LSSM by Shumway and Stoffer (1982).

Specifically for the case of the LSSM, the first E-step starts from the initial values of the parameters $\mathbf{A}, \mathbf{Q}, \mathbf{C}, R$, and estimates the hidden dynamic variable $\vec{x}(t)$ using a Kalman filter. With the following definitions

- $z_{t|t'}$:= the predicted value of a quantity $z(t)$ based on the data $y(1), ..., y(t')$,
- $\Omega_{t|t'}$:= the covariance matrix of the estimated $\vec{x}(t)$, and
- $\Delta_{t|t'}$:= the variance of the prediction errors $(y(t) - y_{t|t'})$,

the equations for the Kalman filter are (Kalman 1960, Gelb 1974, Sorenson 1985, Harvey 1989, Aoki 1990, Bomhoff 1994, Hamilton 1994, Mendel 1995):

$$\begin{aligned}
\Omega_{t|t-1} &= \mathbf{A}\Omega_{t-1|t-1}\mathbf{A}^T + \mathbf{Q} && (12)\\
\Delta_{t|t-1} &= \mathbf{C}\Omega_{t|t-1}\mathbf{C}^T + R && (13)\\
K &= \Omega_{t|t-1}\mathbf{C}^T\Delta_{t|t-1}^{-1} && (14)\\
\Omega_{t|t} &= (1 - K\mathbf{C})\Omega_{t|t-1} && (15)
\end{aligned}$$

$$\begin{aligned}
\vec{x}_{t|t-1} &= \mathbf{A}\vec{x}_{t-1|t-1} && (16)\\
y_{t|t-1} &= \mathbf{C}\vec{x}_{t|t-1} && (17)\\
\vec{x}_{t|t} &= \vec{x}_{t|t-1} + K(y(t) - y_{t|t-1}) \quad . && (18)
\end{aligned}$$

There is a crucial difference between the first four equations and the last three. The first four equations, Eq. (12–15), do not contain the data, they only describe relations between the parameters $\mathbf{A}, \mathbf{Q}, \mathbf{C}, R, \Omega, \Delta$, and K. Their purpose is to find the value of K (the *Kalman gain*) that

subsequently enters Eq. (18). K gives the appropriate weight to the added term originating in the error between the actual observation $y(t)$ and prediction $y_{t|t-1}$.

For true prediction, i.e., when $y(t)$ has not yet been observed, Eq. (16) has to be used for the unobserved state variable, and Eq. (17) for the observable. For model parameter estimation, on the other hand, the entire training data can be used, and an improved estimate of $\vec{x}_{t|N}$ can be obtained by the following three equations (Harvey 1989):

$$\mathbf{B} = \Omega_{t|t}\mathbf{A}^T\Omega_{t+1|t}^{-1} \tag{19}$$

$$\vec{x}_{t|N} = \vec{x}_{t|t} + \mathbf{B}(\vec{x}_{t+1|N} - \mathbf{A}\vec{x}_{t|t}) \tag{20}$$

$$\Omega_{t|N} = \Omega_{t|t} + \mathbf{B}(\Omega_{t+1|N} - \Omega_{t+1|t})\mathbf{B}^T \tag{21}$$

This concludes the E-step.

In the subsequent M-step, the parameters $\mathbf{A}, \mathbf{Q}, \mathbf{C}, R$ are updated; an example of the derivation of the equations can be found in Honerkamp (1993). The iterative model fitting process ends when a convergence criterion is met. This concludes the description of how the model parameters are updated in the M-step.

Once a model has been built, its quality can be evaluated by several different criteria, including:

- **Predictive accuracy**. True out-of-sample predictions are generated using Eq. (17) on a test set that comes after the training period. The accuracy of the predictions can be compared to competing models by different evaluation criteria, such as squared errors or robust errors.
- **Whiteness of the prediction errors**. The model should explain all temporal correlations in the data: a perfect model takes the signal and turns it into white noise. Statistically, the question is whether we can reject (at a certain level of significance) the null hypothesis that the residuals are uncorrelated. Following Brockwell and Davis (1991), we use a Kolmogorov-Smirnov test to determine whether the periodogram of the residuals is consistent with a flat white noise spectrum.
- **Generating data from the model**. The distribution of a certain feature can be derived from realizations of the model and compared with that feature as directly computed from the observed data.

For linear models, two additional criteria are useful:

- **Behavior in the time domain** (relaxation times). The parameters in linear models are related to relaxation times of the corresponding oscillators and relaxators. When the relaxation times are too small (of order of one time step), they usually only fit noise, indicates that the order of the model is too large.
- **Behavior in the frequency domain** (spectrum). The spectrum of the linear process can be computed from the parameters of the estimated models through Eq. (11). Since the spectrum of the model should correspond to the expectation of the periodogram of the data, comparing the spectrum to the periodogram is another important qualitative criterion.

The suggestions listed here are just some of the useful general criteria that will be used in this article. For any specific problem, there are additional, more specific smoke alarms and sanity checks.

4 Applications of state space models to finance

This section discusses two common applications of state space models in financial data, and compares them to our approach. For simplicity of notation, this discussion is written for the case of a scalar $x(t)$:

$$x(t) = ax(t-1) + \epsilon(t) \tag{22}$$

$$y(t) = cx(t) + \eta(t) \quad . \tag{23}$$

The dynamic equation, Eq. (22), is characterized by the single AR[1] coefficient a; $\epsilon(t)$ is the dynamic noise that drives the dynamics. The observation equation, Eq. (23), maps the unobserved state $x(t)$ to the observed variable by scaling it with c. The added observational noise, $\eta(t)$, does not enter the dynamics.

4.1 Smoothing

The first approach splits the variance and results in a smoother series. It can be interpreted as a method for trend estimation. Here, parameter a is not estimated from the data to characterize the dynamics (as in our approach), but rather set to unity. Without loss of generality we can also set c to unity, yielding

$$x(t) - x(t-1) = \epsilon(t) \tag{24}$$

$$y(t) = x(t) + \eta(t) \quad . \tag{25}$$

Eq. (24) interprets $\epsilon(t)$ as the first difference of the series. Reducing the variance of $\epsilon(t)$ by moving some of it onto $\eta(t)$ results in $x(t)$ as a smoothed version of $y(t)$. The variance of the original data $y(t)$ is thus decomposed into observational noise, $\eta(t)$, and a smoother signal, $x(t)$. This can be expressed in a Bayesian framework as a prior on the smoothness of the time series, as discussed by Kitagawa and Gersch (1996). Note that Eq. (24) *resembles Brownian motion.* However, it is not to be interpreted that way here, but as a smoothing constraint for the undisturbed signal instead. The smaller $\epsilon(t)$, the smoother $x(t)$.

This smoothing approach is taken in most state space applications in finance. Bolland and Connor (1996) add to this approach a second non-constant part that is a linear function of the difference of the last two values of the state. This is effectively adding a constraint on the second differences (curvatures) of $x(t)$, in addition to the first differences. Moody and Wu (1996), Moody and Wu (1997a), and Moody and Wu (1997b) use two variations of the simple smoothing model with $a = 1$, and use the term "true price" for the smoothed version of the observed prices.

4.2 Variable parameter AR processes

The second variation of the state space model also uses the state equation to model a slowly varying quantity as in Eq. (24), but the interpretation of the observation equation changes substantially.

The constant c from Eq. (23) is replaced by $y(t-1)$. The equation then becomes

$$y(t) = x(t)y(t-1) + \eta(t) \quad , \tag{26}$$

representing an AR[1] process. $x(t)$ has become an autoregressive parameter that slowly varies with time, and the former observational noise $\eta(t)$ now acts as dynamic noise (Wells 1996), whereas we assume the parameters that characterize the system are constant over time.

4.3 Modeling noisy linear systems

The two cases above do not do justice to the dynamic structure of Eq. (22). In contrast, this article focuses on estimating the full hidden dynamics from the data. This allows us to characterize the process as a linear damped system of relaxators and oscillators, driven by dynamic noise, and observed through a veil of added observational noise.

In the econometric literature, *stochastic volatility models* have been used to describe the dynamic structure of returns, see Shephard (1996) for a recent review. In the notation of the present article, a stochastic volatility model can be expressed as

$$x(t) = a_0 + a_1 x(t-1) + \epsilon(t) \tag{27}$$

$$y(t) = \eta(t)\, \exp(x(t)) \quad . \tag{28}$$

The idea behind using $\exp(x(t))$ is to model the skewed distribution of squared returns found for the empirical data. Parameter estimation in this model is cumbersome due to the log-normal distribution of $\exp(x(t))$. It is usually based on the generalized method of moments, quasi-likelihood estimation or Markov chain Monte Carlo methods. In contrast to stochastic volatility models, we apply a static transformation to the data that will be introduced in the next section in order to make the distribution of squared return approximately normal. This allows us to use as standard maximum likelihood framework for the parameter estimation.

5 Data

This article reports results on the following data sets:

- High frequency DEM/USD foreign exchange rates.[5] We began with eight years of data (through June 29, 1995) spaced apart 30 minutes in ϑ-time (Theta-time). We dropped all points with missing values, and then took every fourth of the remaining points for our analysis, effectively downsampling to two hours in ϑ-time.[6] ϑ-time removes daily and weekly seasonality: time of day with a high mean volatility are expanded, and times of day and weekends with low volatility are contracted (Dacorogna, Gauvreau, Müller, Olsen and Pictet 1996).

[5] We thank Michel Dacorogna (Olsen & Associates, Zurich) for the high frequency DEM/USD exchange rate data.

[6] Whether half-hour or two-hour intervals in ϑ-time are taken does not change the results reported here, since the time scale of the dynamics that we find is two orders of magnitude slower than the sampling interval. However, if we were to have changed the sampling interval by a larger factor, note that Brown (1990) shows for S & P 500 Index futures that the estimated (unconditional) volatility decreases by 13% as the sampling interval is changed from one minute to one hour.

- Daily stock indices. We use two stock indices:
  - Nikkei 225 index (40 years of daily data, through October 15, 1996, 12288 points total),[7]
  - Dow Jones Industrial Average (25 years of daily data, through October 16, 1987, 6252 points total).[8]

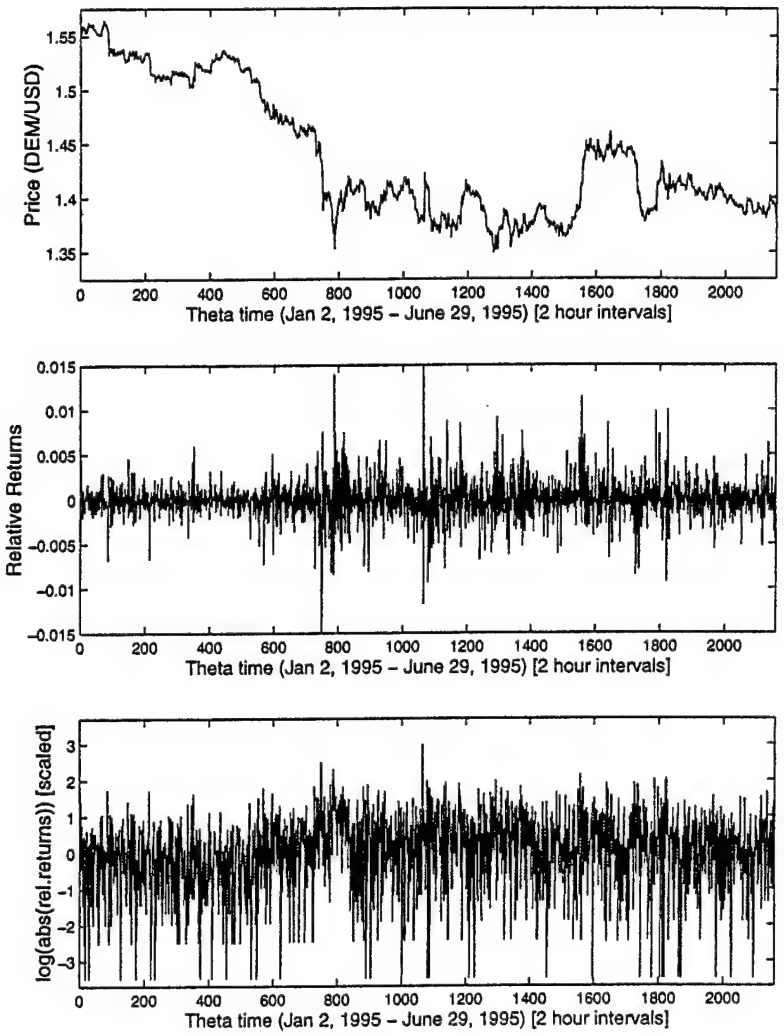

Figure 1: This figure displays a six-month window of the high frequency foreign exchange data, sampled at two hour intervals in ϑ-time. The top panel shows the prices, the middle panel shows the relative returns, and the bottom panel shows the series used in our analysis, i.e., after applying the logarithm and scaling it to zero mean and unit variance.

The top panel of Fig. 1 graphs the level of DEM/USD for the first half of 1995. Its periodogram, shown in the left panel of Fig. 2, drops to first approximation as the spectrum of a random walk whose $1/f^2$ line is also indicated. (f denotes the frequency.) The signature of observational noise—a noise floor masking the signal at high frequencies—is absent: the periodogram continues to drop

[7]We thank Morio Yoda (Nikko Securities, Tokyo) for the Nikkei 225 stock index data.

[8]The Dow Jones Industrial Average data set is described in LeBaron and Weigend (1997) and available through `www.stern.nyu.edu/~aweigend/Research`.

to the highest time scale. The result is that *price levels* $p(t)$ of financial instruments do not exhibit significant observational noise; all the "noise" on prices is dynamic, i.e., it re-enters the dynamic equation.

The central panel of Fig. 1 shows the difference of the logarithm of the price levels

$$\log p(t) - \log p(t-1) = \log \frac{p(t)}{p(t-1)} \approx \frac{p(t) - p(t-1)}{p(t-1)} \quad . \tag{29}$$

This quantity can be interpreted as the logarithm of the geometric growths, i.e., as the logarithm of the ratio of the prices. Using the fact that the logarithm Taylor expands around 1 as $\log \epsilon \approx 1 + \epsilon$, it can also be interpreted as the returns normalized by the levels, i.e., the *relative returns*. Note in the central panel of Fig. 1 that the width of the "band" varies over time; regimes with larger shocks (positive or negative) alternate with regimes with smaller widths. The corresponding periodogram of the relative returns is shown in the right panel of Fig. 2. Note that it is essentially flat: the subsequent returns on the two-hour time scale in ϑ-time appear to be (linearly) uncorrelated.

To exploit this observed structure in the absolute values of the relative returns, we square the relative returns, i.e., ignoring their signs. The distribution of the squared returns is very skewed. To make it less skewed, we take their logarithm,

$$y(t) = \log \left[\log \frac{p(t)}{p(t-1)} \right]^2 \quad . \tag{30}$$

The *logarithm of the squared relative returns*, $y(t)$, is shown for the DEM/USD data in the the bottom panel of Fig. 1.

The squared relative returns can be interpreted as independent realizations of a random variable with a slowly changing mean. If the relative returns $\log p(t)/p(t-1)$ were normally distributed with unit variance, their squares would follow a χ^2_1 distribution. The variance of this χ^2distribution is twice its mean, implying that the realizations are very noisy indeed! This is the source of the observational noise for volatility. On empirical data, it is well known that the relative returns $\log p(t)/p(t-1)$ are not normally distributed, but have fatter tails. However, the spirit of the explanation for the observational noise still applies; see also Diebold and Lopez (1995).

Fig. 3 shows this effect. The periodogram of the data contains most of its power at low frequencies. Subsequently, as the frequency increases, it begins to drop. Finally, it flattens out as the signal gets masked by this "observational noise," stemming from the noisy realizations of the slowly changing means of the squared returns. Note the absence of a daily or weekly peak in this periodogram: while present for data in chronological time, it has been successfully removed by Olsen's projection of the data onto ϑ-time. This periodogram is similar to figures in Schnidrig and Würtz (1995) and in Andersen and Bollerslev (1997). However, neither of these papers interpret the signature as evidence for observational noise, nor do they use a state space model to explain the data.

The key features of the periodogram—a drop over many orders of magnitude for price levels, a roughly constant level for returns, and a low frequency signal disappearing into observational noise at higher frequencies for squared returns—hold for all the financial data sets we analyzed, including six other currencies on different time scales, as well as several stock indices. The next section gives detailed results for DEM/USD and Nikkei 225, as well as brief results for the Dow Jones industrial index.

6 Results

Table 1 summarizes the results for the high frequency DEM/USD data, comparing linear state space models with ordinary AR models. The linear state space models differ crucially from the AR models in the decay times τ: while the decay times of the state space models are significant, they are negligible for the AR models where the processes typically decay within one time step. Since the state space model is fitted to $y(t)$ as defined in Eq. (30), the decay times characterize when the logarithm of the squared relative returns has decayed to 37% of its initial value.

For first order models describing a single relaxator, there is a huge difference in decay time

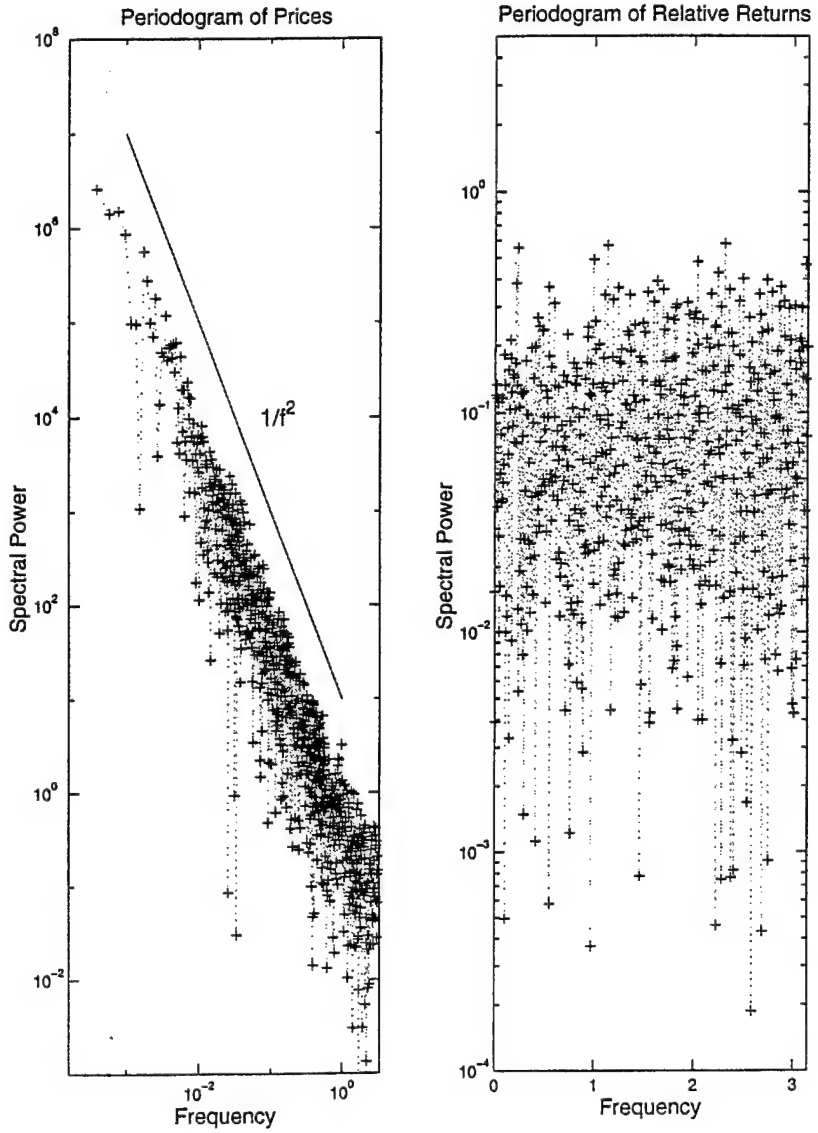

Figure 2: Periodogram of the DEM/USD prices (left), and of the relative returns (right). Expressed in 1/time, the leftmost points correspond to 1/(8 years), 1/(4 years), 1/(2.6 years), 1/(2 years). To guide the eye, we also plotted the $1/f^2$ drop in spectral power of a random walk over six orders of magnitude. The periodogram of the returns on the right hand side is essentially flat. Neither the prices nor the returns indicate the presence of observational noise, in contrast to Fig. 3.

between 156 time steps for the LSSM in contrast to an insignificant 0.45 time steps for the AR process. The eigenvalues of the second order models, given by Eq. (6), turn out to be real; the process thus corresponds to the superposition of two relaxators. The slower one of the two relaxators settles to around 240 of the 2-hour steps and corresponds to 20 days, whereas the slower AR relaxator still decays in a single time step. Using third and fourth order, oscillators emerge whose resonance frequencies $1/T$ correspond to about one day. They might indicate a tiny amount of periodicity left after the transformation of the raw data to ϑ-time, but they do not contribute significantly to the dynamics since their relaxation times are of the order of a few time steps only.

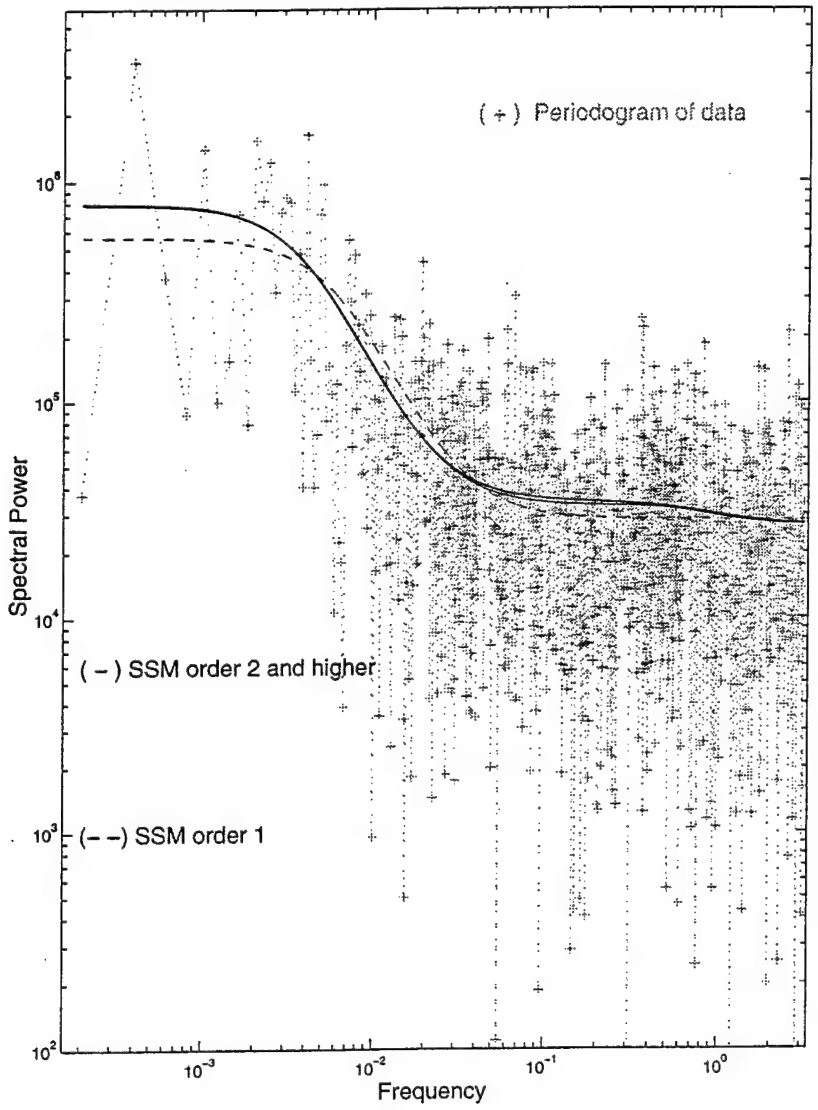

Figure 3: Periodogram ("+") of the DEM/USD exchange rates, and spectra of the estimated state space models (SSM) of order one (dashed line) and higher orders (solid lines).

Models DEM/USD	τ (decay times) 1 step = 2 hours	AR coefficients	Prob white noise	E_{NMS}
$LSSM(1)$	156	0.994	0	0.960
$LSSM(2)$	240, 1.09	$\begin{pmatrix} 0.996 & 0 \\ 0 & 0.399 \end{pmatrix}$	0.72	0.957
$LSSM(3)$	236 $[T = 17\ \tau = 1.6]$	$\begin{pmatrix} 0.966 & 0 & 0 \\ 0 & 0.507 & -0.188 \\ 0 & 0.188 & 0.507 \end{pmatrix}$	0.57	0.957
$LSSM(4)$	243, 1.1 $[T = 8.5\ \tau = 10]$	$\begin{pmatrix} 0.996 & 0 & 0 & 0 \\ 0 & 0.411 & 0 & 0 \\ 0 & 0 & 0.666 & -0.612 \\ 0 & 0 & 0.612 & 0.666 \end{pmatrix}$	0.70	0.957
$AR(1)$	0.45	0.107	0	0.988
$AR(2)$	0.85, 0.64	0.100 0.066	0	0.984
$AR(3)$	1.25 $[T = 6.5\ \tau = 0.86]$	0.097 0.062 0.044	2e-6	0.982
$AR(4)$	1.8, 1.2 $[T = 4.1\ \tau = 1.2]$	0.095 0.058 0.039 0.049	7e-4	0.979

Table 1: Results for the volatilities of the DEM/USD exchange rates. While linear state space models (LSSM) of order two and above fit the data well, ordinary AR models cannot explain the structure of the data.

The decay constants presented here are defined for the logarithm of the squared relative returns. Nonlinear transformations do not allow for an amplitude-independent interpretations of decay times in general. However, fitting state space models directly to the absolute or squared relative returns (without taking the logarithm) yields similar decay constants. This implies that our characterization also hold for stochastic volatility models.

The fourth column in Table 1 shows that the residuals of the state space model of order one are not consistent with white noise, implying that a first order LSSM does not describe the data adequately. However, all higher order LSSMs produce residuals consistent with white noise at a significance level of 0.05 for the Kolmogorov-Smirnov test on the whiteness of the residuals (Brockwell and Davis 1991). None of the residuals of the AR models are consistent with white noise. This is another indication that AR models are not an adequate model class for volatility.

The last column gives the normalized mean squared error, E_{NMS}, between the observed $y(t)$ and the predictions obtained via Eq. (17). Whereas for LSSM, the error drops quickly to a constant level of 0.957 at order 2, it decreases for AR models at a much slower rate, and also remains at a higher level. In an AR(10) model, for example, E_{NMS} takes the value of 0.973, still significantly above the value of the second order LSSM.

We now turn to the power spectra. The curves in Fig. 3 are the power spectra of the state space models.[9] They are computed using Eq. (11). There is a clear difference between the first order spectrum and the higher order spectra. The higher orders ($\geqslant 2$) are very similar, indicating that the second order state space model is indeed sufficient. The spectra of the state space models

[9]The spectra and the periodogram are normalized. For the lowest 200 frequencies, all periodogram points are plotted. Above this frequency, they are logarithmically thinned out for the sole reason to keep the files reasonably small for the on-line version. The visual impression in the printed version does not change.

correspond well to the periodogram of the data. Note that the spectra are not obtained by some direct smoothing of the periodogram in frequency space, but are the spectra of the state space models which were fitted in the time domain.

Models Nikkei225	τ (decay times) 1 time step = 1 day	Prob of white noise	E_{NMS}
$LSSM(1)$	63.1	0.004	0.906
$LSSM(2)$	81.8, 1.45	0.56	0.905
$LSSM(3)$	81.2 $[T = 8.7\ \tau = 6.9]$	0.64	0.905
$LSSM(4)$	81.7, 1.46 $[T = 8.4\ \tau = 10]$	0.57	0.905
$AR(1)$	0.54	0	0.975
$AR(2)$	1.20, 0.82	0	0.959
$AR(3)$	1.85 $[T = 6.6\ \tau = 1.05]$	7e-7	0.951
$AR(4)$	2.93, 1.59 $[T = 4.15\ \tau = 1.61]$	0.002	0.940

Table 2: Results for the volatilities of the Nikkei 225 stock index. While linear state space models of order two and above fit the data well, ordinary AR models cannot explain the structure of the data.

The results for the second data set, the logarithm of absolute values of the relative changes of the daily Nikkei 225 level, are summarized in Table 2. The key point is the large decay time of about 3 1/2 months, revealed by the state space models of order two and above, as well as the failure of AR models, very similar to the DEM/USD data set discussed.

The third data set, the logarithm of absolute values of the relative changes of the daily Dow Jones Industrial Index, reveals a decay time of 117 days or about 5 months. In that case, a one dimensional hidden state already generates residuals that are consistent with white noise. As in the other two examples, no ordinary AR model in the observed variable explains the data. This effect will be clarified in the next section.

7 Ignoring observational noise

The failure of AR models shown in the previous section is a consequence of the observational noise that is present in the volatility data. Whereas linear state space models include the observational noise explicitly in the model, autoregressive models assume that the data is free from observational noise. We use a simple first order process to demonstrate the consequences of ignoring observational noise on the autoregressive parameter.

In an AR[1] model, $x(t) = ax(t-1) + \epsilon(t)$, the parameter a can be estimated without bias as

$$\widehat{a} = \frac{\sum x(t-1)x(t)}{\sum x(t-1)x(t-1)} \quad . \tag{31}$$

If, however, the dynamics is covered by observational noise

$$y(t) = x(t) + \eta(t), \qquad \eta \sim \mathcal{N}(0, R) \quad , \tag{32}$$

the expected value (denoted by $<\cdot>$) of $\widehat{a}$, estimated in analogy to Eq. (31) from $y(t)$, now becomes

$$<\widehat{a}>=\frac{<y(t-1)\quad y(t)>}{<y(t-1)\quad y(t-1)>}=\frac{a}{1+R/<x(t)^2>}\quad . \tag{33}$$

Thus, the larger the variance R of the observational noise, the worse the parameter a will be underestimated. This effect is known from linear regression as the problem of *errors-in-variables* (Fuller 1987). It was first mentioned in time series context by Kostelich (1992), see also König and Timmer (1997). The underestimation of the functional relation between past and present values carries over to more general models, including nonlinear models (Carroll, Ruppert and Stefanski 1995, Weigend, Zimmermann and Neuneier 1996).

8 Summary and Applications

This article showed the important distinction between observational and dynamic noise. When observational noise is present, an autoregressive approach cannot model the data adequately—a state space approach is needed to capture the hidden dynamics. In finance, neither prices nor returns tend to have observational noise. However, volatilities do exhibit signature of observational noise in the periodogram: for low frequencies, there is structure above the noise floor of observational noise.

We showed on three representative financial data sets that a linear state space model with full dynamics can describe volatilities well. We also showed that the resulting models can be nicely interpreted, both from the perspective of physics as a superposition of two simple relaxators, and from the perspective of finance as volatility clustering with a decay time of about three weeks (for DEM/USD), 3 1/2 months (for Nikkei 225), and 5 months (for Dow Jones Industrial Average). These results are in strong contrast to AR models that ignore observational noise and consequently have a bias toward too small coefficients, as shown in Section 7. The more promising modeling approach using state space models over AR models for volatility suggests several applications in financial markets, including

- **Estimating risk.** Knowing the evolution of the volatility is important for determining the risk associated with a position on a financial instrument: the volatility can be interpreted as the conditional standard deviation of the returns.
- **Pricing derivative securities.** Using financial theory, discrepancies between the predicted volatility and the implied volatility can be translated into mispricings, which can in turn be exploited in trading.
- **Information for regime switching models.** The predicted volatility can be an important input for trading models based on the "gated experts" architecture (Weigend, Mangeas and Srivastava 1995). In this case, the hidden state is offered as an additional input to the gate to help determine the current region.

In summary, we discussed the signature of observational noise in the frequency domain and showed on three data sets that volatilities exhibit that signature, but not the prices or returns. We showed that allowing for a hidden process with two or more degrees of freedom, and modeling the

full dynamics of this process, gives interpretable results yielding residuals consistent with white noise. We are currently evaluating on several time horizons the performance for true volatility predictions of state space models in comparison to historic data (Figlewski 1994), GARCH (Bollerslev et al. 1995), and stochastic volatility models (Shephard 1996).

Acknowledgments

Jens Timmer gratefully acknowledges the hospitality of the Information Systems Department of NYU's Leonard N. Stern School of Business. Andreas Weigend thanks Joel Hasbrouck and Steve Figlewski for their comments, as well as the participants of the NBER/NSF Forecasting Seminar for discussions at the NBER 1997 Summer Institute, and acknowledges support from the National Science Foundation (ECS-9309786) and the Air Force Office of Scientific Research (F49620-96-1-0240).

References

Andersen, T. G. and Bollerslev, T. (1997). Heterogeneous information arrivals and return volatility dynamics: Uncovering the long-run in high frequency returns, *Journal of Finance* **52**: 975–1005.

Aoki, M. (1990). *State Space Modeling of Time Series*, Springer-Verlag, New York.

Bengio, Y. and Frasconi, P. (1995). An input output HMM architecture, *in* G. Tesauro, D. S. Touretzky and T. K. Leen (eds), *Advances in Neural Information Processing Systems 7 (NIPS*94)*, MIT Press, Cambridge, MA, pp. 427–434.

Bolland, P. J. and Connor, J. T. (1996). Identification of FX arbitrage opporunities with a non-linear multivariate Kalman filter, *in* A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend (eds), *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets (NNCM-95)*, World Scientific, Singapore, pp. 122–134.

Bollerslev, T., Engle, R. F. and Nelson, D. B. (1995). ARCH models, *in* R. F. Engle and D. L. McFadden (eds), *Handbook of Econometrics*, Vol. 4, North-Holland, New York, NY, chapter 49.

Bomhoff, E. J. (1994). *Financial Forecasting for Business and Economics*, Academic Press, London.

Brockwell, P. J. and Davis, R. A. (1991). *Time Series: Theory and Methods*, Springer-Verlag, New York.

Brown, S. (1990). Estimating volatility, *in* S. Figlewski, W. L. Silber and M. G. Subrahmanyam (eds), *Financial Options: From Theory to Practice*, Business One Irwin, Homewood, IL, pp. 516–537.

Carroll, R., Ruppert, D. and Stefanski, L. (1995). *Measurement Error in Nonlinear Models*, Chapman and Hall, London.

Casdagli, M. (1989). Nonlinear prediction of chaotic time series, *Physica D* **35**: 335 – 356.

Dacorogna, M. M., Gauvreau, C. L., Müller, U. A., Olsen, R. B. and Pictet, O. V. (1996). Changing time scale for short-term forecasting in financial markets, *Journal of Forecasting* **15**: 203–227.

Dempster, A., Laird, N. and Rubin, D. (1977). Maximum likelihood from incomplete data via EM algorithm, *J. Roy. Stat. Soc.* **39**: 1–38.

Diebold, F. X. and Lopez, J. A. (1995). Modeling volatility dynamics, *in* K. Hoover (ed.), *Macroeconometrics: Developments, Tensions, and Prospects*, Kluwer Academic Press, Boston, pp. 427–472.

Engle, R. F. (ed.) (1995). *ARCH: Selected Readings*, Oxford University Press, Oxford, UK.

Figlewski, S. (1994). Forecasting volatility using historical data, *Working Paper S-94-13*, Salomon Center, Leonard N. Stern School of Business, New York University.

Fraser, A. M. and Dimitriadis, A. (1994). Forecasting probability densities by using hidden Markov models, *in* A. S. Weigend and N. A. Gershenfeld (eds), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Reading, MA, pp. 265–282.

Fuller, W. A. (1987). *Measurement Error Models*, John Wiley, New York.

Gelb, A. (1974). *Applied Optimal Estimation*, MIT Press, Cambridge, MA.

Hamilton, J. D. (1994). *Time Series Analysis*, Princeton University Press, Princeton.

Harvey, A. C. (1989). *Forecasting Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge.

Honerkamp, J. (1993). *Stochastical Dynamical Systems*, VCH, New York.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *Trans. ASME J. Basic Eng. Series D* **82**: 35–45.

Kitagawa, G. and Gersch, W. (1996). *Smooothness Priors Analysis of Time Series*, Springer-Verlag, New York.

König, M. and Timmer, J. (1997). Analyzing X-ray variability by linear state space models, *Astronomy and Astrophysics Suppl. Ser.* **124**: 589–596.

Kostelich, E. (1992). Problems in estimating dynamics from data, *Physica D* **58**: 138.

Lapedes, A. and Farber, R. (1987). Nonlinear signal processing using neural networks, *Working Paper LA-UR-87-2662*, Los Alamos National Laboratory, Los Alamos, NM.

LeBaron, B. and Weigend, A. S. (1997). A bootstrap evaluation of the effect of data splitting on financial time series, *IEEE Transactions on Neural Networks* **8**: forthcoming.

Mendel, J. M. (1995). *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, Prentice Hall, New Jersey.

Moody, J. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units, *Neural Computation* **1**: 281–294.

Moody, J. E. and Wu, L. (1996). What is the "true price"? – State space models for high frequency financial data, *Progress in Neural Information Processing (ICONIP'96)*, Springer, Berlin, pp. 697–704.

Moody, J. E. and Wu, L. (1997a). What is the "true price"? – State space models for high frequency FX data, *in* A. S. Weigend, Y. S. Abu-Mostafa and A.-P. N. Refenes (eds), *Decision Technologies for Financial Engineering: Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets (NNCM-96)*, World Scientific, Singapore, pp. 346–358.

Moody, J. E. and Wu, L. (1997b). What is the "true price"? – State space models for high frequency FX data, *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, IEEE Service Center, Piscataway, NJ, pp. 244–252.

Oppenheim, A. V. and Schafer, R. W. (1989). *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ.

Phadke, M. and Wu, S. (1974). Modeling of continuous stochastic processes from discrete observations with application to sunspots data, *J. Am. Stat. Ass.* **69**: 325–329.

Poggio, T. and Girosi, F. (1990). Networks for Approximation and Learning, *Proceedings of the IEEE* **78**: 1481–1497.

Priestley, M. (1981). *Spectral Analysis and Time Series*, Academic Press, London.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE* **77**: 257–286.

Schnidrig, R. and Würtz, D. (1995). Investigation of the volatility and autocorrelation function of the USD/DEM exchange rate on operational time scales, *Proceedings of the First International Conference on High Frequency Data in Finance (HFDF-I)*, Vol. 3, Zurich.

Shephard, N. (1996). Statistical aspects of ARCH and stochastic volatility, *in* D. R. Cox, D. V. Hinkley and O. E. Barndorff-Nielsen (eds), *Time Series Models In Econometrics, Finance and Other Fields*, Chapman and Hall, London, pp. 1–67.

Shi, S. and Weigend, A. S. (1997). Taking time seriously: Hidden Markov experts applied to financial engineering, *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, IEEE Service Center, Piscataway, NJ, pp. 244–252.

Shumway, R. and Stoffer, D. (1982). An approach to time series smoothing and forecasting using the EM algorithm, *J. Time Ser. Anal.* **3**: 253–264.

Sorenson, H. W. (1985). *Kalman Filtering: Theory and Application*, IEEE Press.

Tjostheim, D. and Auestad, B. (1994). Nonparametric identification of nonlinear time series: Selecting significant lags, *J. Am. Stat. Ass.* **89**: 1410–1419.

Weigend, A. S., Huberman, B. A. and Rumelhart, D. E. (1990). Predicting the future: A connectionist approach, *International Journal of Neural Systems* **1**: 193–209.

Weigend, A. S., Mangeas, M. and Srivastava, A. N. (1995). Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting, *International Journal of Neural Systems* **6**: 373–399.

Weigend, A. S., Zimmermann, H. G. and Neuneier, R. (1996). Clearning, *in* A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend (eds), *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets (NNCM-95)*, World Scientific, Singapore, pp. 511–522.

Wells, C. (1996). *The Kalman Filter in Finance*, Kluwer Academic Publishers, Dordrecht.

Yule, G. (1927). On a method of investigating periodicity in disturbed series with special reference to Wolfer's sunspot numbers, *Phil. Trans. Roy. Soc. London* **A 226**: 267–298.

[3]

A Bootstrap Evaluation of the Effect of Data Splitting on Financial Time Series

Blake LeBaron and Andreas S. Weigend

Abstract— **This article exposes problems of the commonly used technique of splitting the available data into training, validation, and test sets that are held fixed, warns about drawing too strong conclusions from such static splits, and shows potential pitfalls of ignoring variability across splits. Using a bootstrap or resampling method, we compare the uncertainty in the solution stemming from the data splitting with neural network specific uncertainties (parameter initialization, choice of number of hidden units, etc.). We present two results on data from the New York Stock Exchange. First, the variation due to different resamplings is significantly larger than the variation due to different network conditions. This result implies that it is important to not over-interpret a model (or an ensemble of models) estimated on one specific split of the data. Second, on each split, the neural network solution with early stopping is very close to a linear model; no significant nonlinearities are extracted.**

Keywords—**Model evaluation. Model uncertainty. Bootstrap. Resampling. Financial forecasting. Time series prediction. Linear bias of early stopping. Superposition of forecasts. Model merging.**

Data—**Dow Jones Industrial Average, 1962-1987. Volume from New York Stock Exchange, 1962-1987. The data used in this article is available from the web sites of the authors.**

I. Introduction

Training a network on a time series is not hard, but once we have a network, how much can we trust the forecasts for truly new data? On the one hand, if the time series is fairly long (above a few thousand points), and if it is fairly clean (noise of less than one percent of the signal), the evaluation of a model is relatively easy, since only very few functions will fit some held-back data very well. This regime can be described as a "right-with-probability-$(1-\epsilon)$-regime." On the other hand, for very noisy and/or very short time series, one can only hope to be right on new data with a probability of $(0.5+\epsilon)$. An example would be the forecast of the direction of a stock price movement. It is well known that random predictions, or random trading strategies, can yield deceptively long sequences of good predictions or profitable trades. In such noisy problems, many functions will be indistinguishable in their forecasting quality. When connectionist techniques are used, additional choices (such as the architecture, training procedure, and the random initialization of the network) make the evaluation even harder. Evaluating a model for noisy time series can be more work than estimating the parameters.

A standard procedure for evaluating the performance of a model is to split the data into one training set (used for the parameter estimation, e.g., through gradient descent or second order methods), one validation set (used to determine the stopping point before overfitting occurs, and/or used to set additional parameters or hyperparameters, such as the importance given to penalize model complexity), and one or more test sets. This procedure has been used for many years in the connectionist community, see e.g., Weigend *et al.* (1990). Our more recent experience has found this approach, along with conclusions drawn from it, to be very sensitive to the specific splitting of the data. Therefore, usual tests of forecast reliability can easily be overly optimistic.

This article addresses these problems with a bootstrap method. The approach we present combines the purity of splitting the data into three disjoint sets with the power of a resampling procedure, giving a better statistical picture of forecast variability, including the ability to estimate the effect of the randomness of the splits of the data vs. the randomness of initial conditions of the network.

This is not the first article that uses the bootstrap in a connectionist context. Weigend *et al.* (1992) used the *bootstrapping of residuals* to evaluate the forecasting power of a neural net for exchange rate forecasts, and Connor (1993) also bootstrapped residuals to obtain error bars for the iterated time series predictions. The goals were different from the goal of the work reported here. In this article we *resample pairs* which will be clarified in Section II-A. Resampling pairs was first suggested by Efron (1982), and first used in the connectionist community by Paass (1993) on the example of noisy exclusive OR. Tibshirani (1996) applied the bootstrap machinery to networks in a cross-sectional context. However, none of these articles evaluate the effect of using the common, simple, static sample split on the performance reliability.

To demonstrate our method, we wanted to use a data set that lies somewhere between simple noise-free function fitting, and a sequence of true random numbers where no model has a chance. We picked the daily trading volume[1] on the New York Stock Exchange, where predictions can explain about half of the variance. Section II of this article describes the method and the data set, Section III presents the empirical results of the study, Section IV discusses other sources of uncertainty not captured by the bootstrap, and Section V draws some conclusions.

The authors can be reached at:
Blake LeBaron, Department of Economics, University of Wisconsin, 1180 Observatory Drive, Madison, WI 53713, blebaron@facstaff.wisc.edu, http://www.econ.wisc.edu/~blake; **Andreas Weigend**, Department of Information Systems, Leonard N. Stern School of Business, New York University, 44 West 4th Street, MEC 9-74, New York, NY 10012, aweigend@stern.nyu.edu, http://www.stern.nyu.edu/~aweigend .

[1] Although forecasting prices is a potentially more lucrative target, volume actually is interesting to the economist whose goal is to understand how markets function.

II. Experimental Design

A. Bootstrapping Methodology

Randomness enters naturally in two ways in neural network modeling: in the splitting of the data, and in choices about the network initialization, architecture, and training. A standard procedure for finding a good network is to split the patterns derived from a time series into three sets: training, validation, and test sets. The training set is used for parameter estimation (in simple backpropagation, by updating the parameter by gradient descent on some cost function). In order to avoid overfitting, a common procedure is to use a network sufficiently large for the task, to monitor (during training) the performance on the separate validation set, and finally to choose the network that corresponds to the minimum on the validation set, and employ it for future purposes such as the evaluation on the test set. These sets have no patterns in common. The usual procedure fixes these sets. As many statistical quantities as desired can be estimated in the test set, but this leaves one question wide open: *What is the variation in performance as we vary training, validation, and test sets?* This is an important question since real world problems don't come with a tag at each pattern saying how it should be used! Also, if we were to only train one network on such a split, this would not tell us how stable the performance is with respect to network choices.

Since there is not just one "best" split of the data or obvious choice for the initial weights etc., we will vary both the data partitions and network parameters in order to find out more about the distributions of forecast errors. We use a computer intensive bootstrapping method to evaluate the performance, reliability and robustness of the connectionist approach, and to compare it with linear modeling. Bootstrapping involves generating empirical distributions for statistics of interest through random resampling. We combine bootstrapping along with random network selection and initialization.

In more detail, in order to understand the impact of the splitting and network choices, we draw a realization of splits and network conditions, and train a complete model on this realization. This is sometimes called bootstrapping *pairs* (Efron & Tibshirani, 1993), since the input-output pairs or patterns remain intact, and are resampled as full patterns. This can be contrasted with training one model only, and resampling the errors of that one model to obtain a distribution, called bootstrapping *residuals*. The latter method was used in single-step prediction by Weigend *et al.* (1992) in the context of foreign exchange rate predictions. One model was built on one split of the data. Similarly, in an application to load forecasting, Connor (1993) trained one single-step prediction network on one split of the data, then resamples from the empirical distribution of the single-step errors and adds these to the inputs in order to obtain estimates of the errors of iterated forecasts. In this residuals bootstrap, the residuals obtained from one specific model are used in rebuilding pairs or patterns to obtain error bars reflecting all sources of error, including model misspecification. In contrast, here we are interested in variation due to sample splits rather than error bars. Every "run" has a different assignment between the sample patterns and the three sets which thus are different for each run.

In the example used in this article, we have some 6200 patterns, each made up of a few past values of a number of time series (for details of how the patterns are constructed, see Section II-B below). We first build the test set by randomly picking 1500 patterns with replacement. The patterns used in this specific test set are then removed from the pool. From the remaining patterns, we then randomly set aside 1500 patterns as the validation set (these are picked without replacement, and are also removed from the pool). The remaining patterns then constitute the training set.[2] For the results presented in the article, we do this 2523 times, training a network each time.

We use fully connected feedforward networks with one hidden layer of tanh units and a linear output unit. However, in order to include variations over reasonable choices for network and learning parameters, a number of network characteristics are also drawn randomly at the beginning of each run.[3] The cost function is the squared difference between the network output and the target (expressed as the log-transformed volume, detailed in the next section), summed over all patterns in the training set. Most results are given in terms of $(1 - R^2)$, i.e., one minus the squared correlation coefficient between forecast and target.

B. Data Set

We use daily data from the New York Stock Exchange (NYSE) from December 3rd, 1962, through September 16th, 1987, corresponding to 6230 days.[4] Our forecasting goal is daily total trading volume, shown in Fig. 1. We believe that this series has two interesting features: First, while many articles have tried neural network approaches to forecasting *prices*, few have attempted forecasting trading *volume*. Second, volume differs from many other financial series in that it contains more forecastable structure than typical price series. We use the daily measure of aggregate turnover on the NYSE which is total volume

[2]There is no deep theoretical justification for drawing the test data with replacement, and the training and validation set effectively without replacement. Our motivation was to stick to the standard rule of sampling with replacement for the test set. For the training and validation sets, we did not allow for repeated patterns since we wanted the linear fit comparison to be estimated on each non-test data point with even weight, and wanted to use identical sets for the net and the linear fit in each run.

[3]In detail, the network architecture is chosen uniformly over 2 to 6 hidden units. The learning rate is chosen uniformly over $[1, 20] \times 10^{-4}$, no momentum. The weight-range w of the initial weights is drawn between $[0.25, 2.5]$. The individual weights are then initialized randomly from a uniform distribution over $[w/i, -w/i]$ where i is the number of connections coming into a unit ("fan-in"). The block-size (how many patterns are presented until the weights are updated) is drawn uniformly from $[20, 180]$. All inputs are scaled to have zero mean and unit variance as estimated over the entire data set. No significant correlation was found between performance and any of these choices.

[4]A "super test set" (the period from September 17th through October 19th, 1987 that contains the 1987 crash) is set aside for some final out-of-sample forecasting experiments, described in Section III-D.

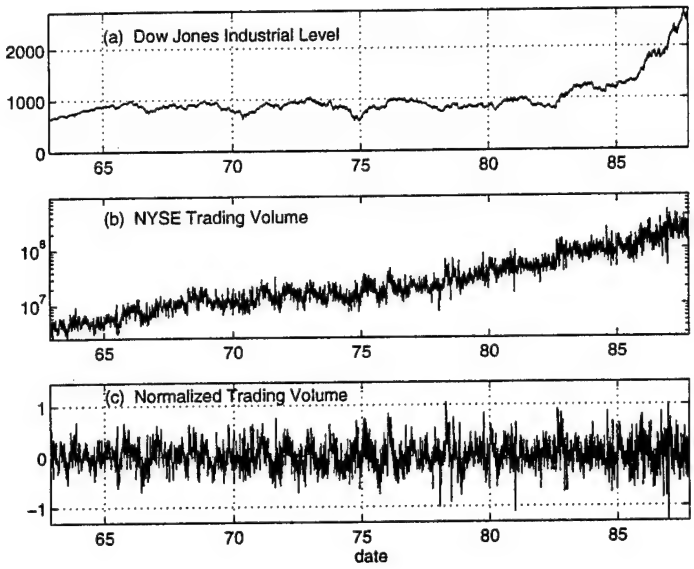

Fig. 1. (a) The level of the Dow Jones Industrial Average from December 1962 - October 1987. (b) The raw trading volume (aggregate turnover) on the NYSE. The nonstationarity is evident; this is a semi-logarithmic plot. (c) The series v_t that we use as target: it is obtained by taking the logarithm of the raw value and dividing it by the mean of the last 100 trading days.

divided by shares outstanding, or the fraction of shares traded that day. This series is not stationary, Fig. 1 (b). We "detrend" it by dividing by a 100-day moving average of past turnover. In other words, we compare the volume today with the average volume over the last 100 trading days. The distribution of this series is still very skewed. We then take the logarithm to obtain a less skewed distribution.[5] We refer to this transformed series as v_t. This target series is shown in Fig. 1 (c).

Beside three lagged values of v (a typical autoregressive or AR model), we use three other sets of variables, making it an exogenous or ARX model. We use first differences of the logarithm of the level of the Dow Jones Industrials Index as a measure of relative stock returns, r_t. Furthermore, volume movements are connected to stock return movements in interesting ways (Karpov, 1987; LeBaron, 1992a; Gallant *et al.*, 1993). One of these features is that volume is related to stock price volatility, sometimes approximated by the absolute magnitude of daily price movements. Furthermore, volume tends to be higher in rising markets. For these reasons we chose several lagged returns and volume variables as predictors. The predictor vector (i.e., the 12 values presented to the network as inputs for each pattern) is given by

$$\{\ v_{t-1,2,3}\ ,\ r_{t-1,2,3}\ ,\ |r_{t-1,2,3}|\ ,\ \log(\sigma^2_{t-1,2,3})\ \}\ .$$

[5]Normalizing with the 250-day mean (of the last trading year) did not remove quite enough of the nonstationarity. Note also that we are using the normalized level of the volume, not a difference version that would correspond to the change in volume. Apart from correcting somewhat for the skewed distribution, the logarithm can be interpreted as emphasizing small values of the volume more than large ones, and, alternatively, as facilitating product interactions between lagged values of the volume, since the inputs are added in the argument of the hidden units, and adding logarithms corresponds to multiplying the original values.

Here, σ_t is an estimate of a volatility. It is defined recursively as

$$\sigma_t^2 = \beta\ \sigma_{t-1}^2 + (1-\beta)\ r_t^2 \qquad \text{with } \beta = 0.9\ .$$

This represents an exponential filter of the squared returns. This can be interpreted in physical terms as a relaxator: A shock in r_t^2 decays in $-1/\log\beta = 9.5$ days to $1/e = 0.37$ times its initial value.[6] We initialize σ_0^2 to the unconditional variance of the series. The choice of the exponentially smoothed squared returns is motivated by the similarity to variance estimates from autoregressive conditional heteroskedastic (ARCH) models often used in financial time series (Bollerslev *et al.*, 1990; Bollerslev *et al.*, 1995).

Summarizing, we use the following inputs for our model:

- Three lags of the past trading volume, $v_{t-1,2,3}$. They are normalized by the 100-day moving average (but not differenced), see Fig. 1 (c). Their one-day autocorrelation after normalization is 0.66. (Without our normalization, i.e., taking the raw volume from Fig. 1 (b), the overall shift in level over the two decades is responsible for an autocorrelation of 0.95.)
- Three lags each of the relative returns, $r_{t-1,2,3}$. Their one-day autocorrelation is small (0.135), and disappears for two or more lags, as discussed in LeBaron (1992).
- Two estimates of their volatilities, with three lags each:
  - Absolute value of the relative returns,$|r_{t-1,2,3}|$. Their autocorrelation coefficients are dropping off very slowly, and have values for the first 10 lags around 0.16, computed after subtracting the mean of $|r_t|$.[7]
  - Logarithm of the exponentially smoothed squared returns, $\log(\sigma^2_{t-1,2,3})$. Their one-day autocorrelation is 0.975. It drops off very slowly, primarily due to the smoothing (each value re-enters at the next time step attenuated by $\beta = 0.9$), and secondarily due to the already existing autocorrelation of the driving process of r_t^2.

We refer to each of these 12-dimensional predictor vectors with the associated 1-dimensional target value as *a pattern.* The correlation coefficients were computed through Oct 19, 1987, i.e., excluding the effect the day of the 1987 crash would have. As shown, some of the input dimensions are highly correlated. Despite this high correlation that gives an effective overlap of the patterns in the three sets, we will see in the next section that the performance varies

[6]An equivalent box-cart moving average would average the squared returns over 19 days. We chose the exponential average since it does not exhibit the box cart's shadows, i.e., the effect that large shocks show up again with the opposite sign once they drop off the left side of the window.

[7]The 1/e decay time of the corresponding AR process is about half a time step. This does not characterize the time scale of the underlying process well: the coefficient is severely underestimated due to the presence of noise in the inputs ("errors-in-variables", see Fuller (1987) and Carroll *et al.* (1995)). Fitting a state space model with full dynamics to the series $\{|r_t| - < |r_t| >\}$ gives an autoregressive coefficient for the dynamics of the state of 0.9915, corresponding to an 1/e decay time of approximately 5 months (117 trading days). For more details of this method and their interpretation for modeling volatilities see Timmer & Weigend (1997).

a lot for different random samples out of these overlapping patterns.

III. Empirical Results

A. Learning Curves and Overfitting

Fig. 2 shows the set of learning curves[8] for a typical run, for the three sets, both expressed as one minus the correlation coefficient squared, $(1 - R^2)$, and as the mean squared error divided by the overall variance of the target, NMSE. Differences between these two reasonable performance measures occur when the mean and the variances are not estimated correctly. Whereas the correlation coefficient corrects for these differences (by subtracting the means and dividing by the standard deviations), the squared error does not, and is thus higher than $(1 - R^2)$.

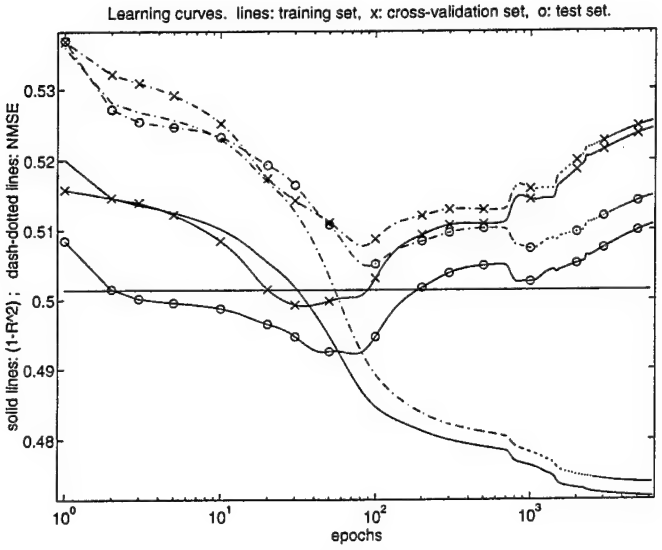

Fig. 2. Learning curves of one specific network on one specific split. They show the performance vs. the number of backpropagation iterations. There are three pairs of curves. The first pair (monotonically decreasing) gives the performance on the training set, the second pair (denoted by "x") on the validation set, and the third pair ("o") on the test set. The three solid lines plot the $(1 - R^2)$ measure; the three dash-dotted lines give the normalized mean squared error (NMSE). The straight line indicates the test error of a linear model estimated on the union of the training set and the validation set.

The learning curves in Fig. 2 show performance vs. the number of backpropagation iterations. There is a clear increase of validation and test errors after passing through minima, usually called overtraining or overfitting. At some stage (around epoch 800 in this specific run which happened to have a very small learning rate) the network extracts a feature of the training set that helps the test set, but hurts the validation set. The minima of the validation set and the test set do not occur at the same epoch. From each of these sets of learning curves, only a single number is used for the subsequent analysis and comparisons in this article: the performance value on the test set at that epoch that has the minimum of the validation set.

[8]We use the term learning curve to characterize the performance as a function of the iterations of the algorithm. In a different context, typically when an arbitrary number of training patterns can be generated, the term learning curve denotes performance as a function of data set size.

B. Linear vs. Nonlinear Comparison

One of the most important goals of any exploration of a nonlinear forecasting method is to demonstrate an improvement over linear forecasts. For synthetic data, generated from nonlinear noise-free systems, forecast improvements of several orders of magnitude have been reported: consider the celebrated logistic map which consists only of a second-order component (quadratic term) without any first-order (linear) component. It really should come as no surprise that methods that allow for nonlinearities will vastly outperform the perfectly inadequate linear fit in cases when there is no linear component.

We here focus on high-noise real world data where the evaluation is much harder, and potential nonlinearities are often masked by noise. In this case, great care needs to be taken to evaluate the nonlinearity of the model: obtained on a single split, depending on the split, a network can easily be a few percent better, but also a few percent worse than the linear model. Thus, instead of just comparing forecasts on one split and one out-of-sample time period, we recommend bootstrapping and reporting the distribution of forecast performance for both the network and linear forecasts. This allows a more meaningful statistical comparison between linear and neural network models.

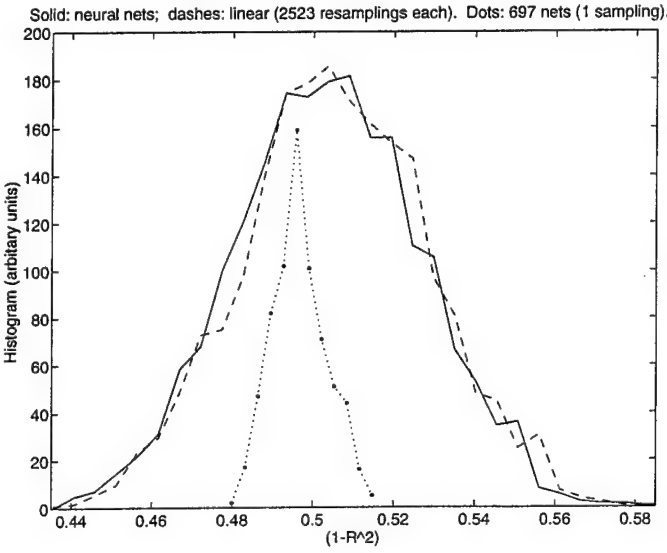

Fig. 3. Histograms of $(1 - R^2)$ forecast performance. The solid line shows the distribution of the networks, the dashed lines of linear model, both estimated on 2523 different resamplings of the available data. The dotted line takes just one split of the data and describes the distribution of 697 networks. The fact that the width of the dotted histogram is clearly smaller than the width of the other two indicates that the randomness in the splitting of the data generates more variability than the randomness in network initialization does.

In this comparison we fit for each split a linear model to exactly the same patterns (inputs and targets) used for the network. Parameters are estimated using the union of the

same training and validation set, and $(1-R^2)$ is estimated over the same test set from each bootstrap resampling.[9]

The empirical density from 2523 bootstrap resamplings of the forecast performance is shown in Fig. 3. The solid line displays the performance of the networks, and the dashes that of the linear models. It is clear from this picture that distinguishing between the two forecasts is going to be difficult, if possible at all.

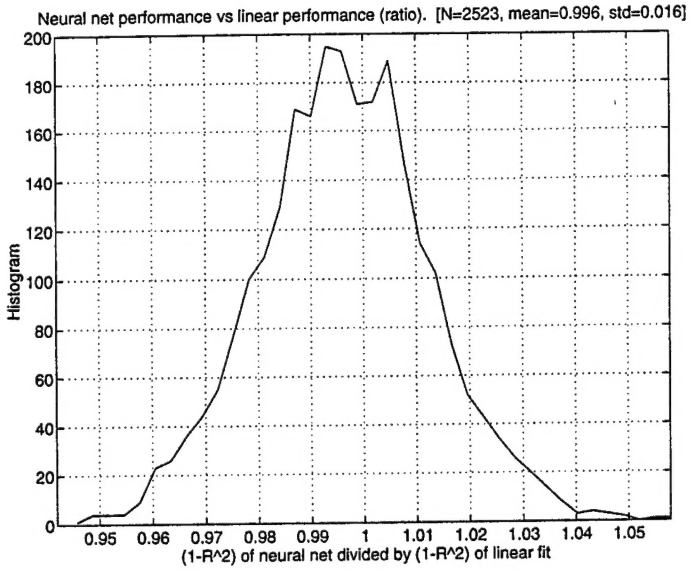

Fig. 4. Histogram of the ratio of $(1-R^2)$ network performance divided by $(1-R^2)$-linear performance for 2523 resamplings. Each entry in this histogram corresponds to the performance ratio of one network and one linear model trained on one specific split.

To focus on the comparison, Fig. 4 shows a histogram of the run-by-run ratio of the two forecast performance measures. This ratio is estimated for each of the bootstrap samples and recorded. If the networks were consistently outperforming the linear models then this ratio would be less than unity. However, this histogram shows that it is not very likely that the network will do better than a linear model in most cases.[10]

Another perspective on the correlation of forecast performance between neural networks and linear models is given in Fig. 5, a scatter plot of the performance of the nonlinear vs. the nonlinear model,

$$\left\{ (1-R_L^2)^i, (1-R_{NN}^2)^i \right\}.$$

[9]One referee suggested a comparison with an autoregressive moving average (ARMA) models instead of the exogenous autoregressive linear (ARX) we use. ARMA models are indeed more general linear models than AR models. However, for the pairs-bootstrap study presented here, where resampling destroys the sequence of the patterns, it is not possible to feed back errors (the MA part). An ARMA model cannot be used in combination with the pairs-bootstrap; ARX is thus the appropriate linear model class to compare to.

[10]The average ratio in Fig. 4 is 0.996 ± 0.016. On the one hand, this is significantly different from 1, with a t-statistic of 12.6, indicating a significant, but small improvement in overall forecast performance. On the other hand, when we compare the forecast performance using squared forecast errors, we find that the average ratio (over the same 2523 runs) is larger than unity, 1.003 ± 0.020 (the confidence intervals are statistical errors of one standard deviation). This leads us to the conclusion that there is no relevant difference between the nets and the linear fits.

One point is entered for each bootstrap sample, i. If the networks are picking up much of the same structure as the linear forecasts, we will see a strong correlation between the two. This is indeed the case in Fig. 5 where the correlation between forecast errors is 0.936.

To summarize this section: When we embarked on this experiment, we were hoping for simple clean evidence for nonlinear structure in the volume of the NYSE, of high interest for economists. What we found instead is that possible underlying nonlinearities are not easily discovered—using a model class celebrated for its ability to express any nonlinear function (feedforward networks with tanh hidden units with a squared error cost function) did not reveal such structure. Since this article focuses on the variation due to different splitting of the data, we did not use explore alternatives to early stopping that avoid the bias towards a linear solution, such as weight-elimination or pruning; those are interesting experiments and the data is available from the authors' web sites. Furthermore, we did not use computer generated nonlinear data, since generating an arbitrarily large number of noise-free data points of an ergodic system will typically (for any split of the data) give very close neighbors between the different sets. This does not constitute a serious test for the real-world problem of noisy data of finite record length, perhaps slightly nonlinear, that we typically find in economics, finance and business.

C. Variability Over Random Networks

Our procedure randomizes both over data samples and over network architectures and initial parameters. An important question is: *How much of the variability is due to the data set resampling, and how much is due to the network parameters?* Viewed from a different angle: for a given split, how much model overfitting would connectionists be likely to engage in were they to optimize their network architecture etc. for that split? If great gains were possible by tinkering with network parameters for each split, we should observe a lot of variability in forecast performance over randomly initialized networks on a given data set split. However, the dotted line in Fig. 3 shows a representative density for 697 randomly drawn nets, all trained and tested using the same training, validation, and test sets. The answer to the question is: *The variations of the forecasts due to changes in network structure are small relative to the variations due to sample splitting.*

D. Probability Density of the Forecasts

Now that we have an entire ensemble of neural network predictors, we can investigate how all these networks can give us a fresh view on the old idea of combining forecasts by looking at the scale of the variations compared to the noise inherent in the problem. We use each of the networks to make predictions on a sample that had been set aside throughout (i.e., never used during training, validating or testing). The time period of this sample starts immediately after the time period considered so far, i.e., it starts on September 17th, 1987, and includes the crash of October 19th, 1987, a day with unusually large price movement

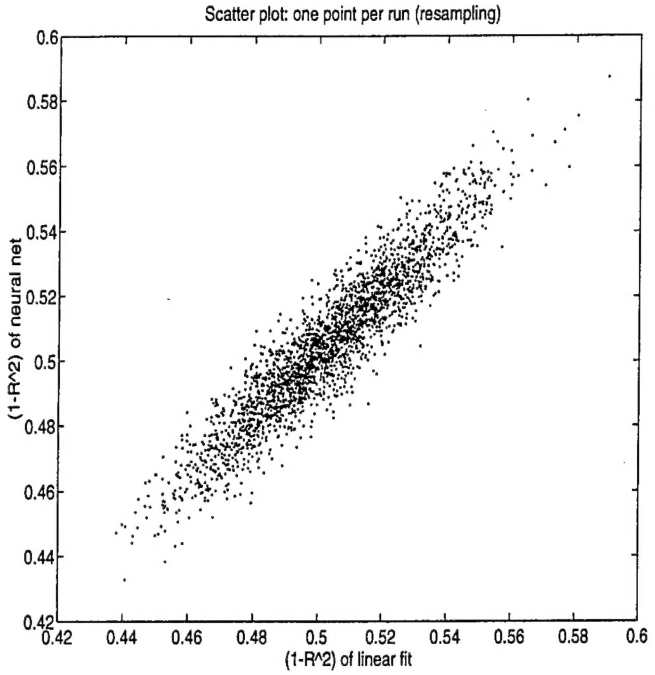

Fig. 5. Scatter plot of the prediction errors. For each of the 2523 runs (i.e., different splits of the data) one point is entered into this scatter plot. Its location is determined by the performance of the network vs. the linear model. Note that the point cloud is much more stretched out along the 45-degree line than orthogonal to it, again indicating that there is more variation due to the randomness in data splits than the variation that results from the randomness in the initial conditions.

and trading volume.

Fig. 6 displays, for each day, the density of all the network predictions. They are single-step forecasts: the input for tomorrow's prediction contains today's observed values. The solid lines are the histograms of the individual raw predictions; they have not been convolved with any added uncertainties. The actual data points are marked with $\times$. (The data points for the stock market crash, October 19th and 20th, 1989 are missing since they are off the scale.) A few interesting features are contained in the figure. First, we see that the forecasts in many cases are biased high or low, indicating generally mediocre forecast performance. (Explaining 50 percent of the variance of the data means that there still remains 50 percent unexplained!) Second, the fact that for many of the days the width of the distribution is quite small and quite far away from the actual value suggests that even the smartest selection or combination of forecasts cannot yield much improvement.[11] Finally, we can see how the models' predictions begin to spread apart as the period of the crash is reached. The main reason for this spreading is that the inputs wander off into regions where the network has never seen training points. Regression neural networks do not spend any resources on modeling the density of the inputs—moving away from region of interpolation to extrapolation manifests itself indirectly through deteriorating performance. Thanks to the benevolent nature of tanh hidden units, the output remains bounded even for thus far unexplored regions.

[11]The idea of combining of forecasts (Bates & Granger, 1969), based on the idea that superposition helps to the degree that the errors are uncorrelated, has recently reached the connectionist community, see e.g., (Jacobs, 1995). This article presents, on a practical example, the limitations of averaging for noisy data: the empirical densities show that averaging over all the splits we did (by taking the mean, median or any convex, possibly even adaptive, combination of the 1843 individual models) will not improve the predictions dramatically.

IV. Relation to other sources of uncertainty

Forecast uncertainty can come from many sources. We focused on the uncertainty obtained from the specific splits, that can be called splitting uncertainty. In the larger picture, its size is relatively small compared to all the noise sources that contribute to the normalized squared error of about 50 per cent, or a correlation coefficient of about

$$R = \sqrt{(1 - 0.5)} = 0.7 \ .$$

We here briefly describe the effect of other sources of uncertainty:[12]

- Noisy targets. An appropriately trained network outputs expected values. Gradient descent in a squared error cost function can be interpreted in a maximum likelihood framework as the observed values being normal distributed around the predicted values with constant noise level. This assumption can be relaxed, first by allowing a Gaussian with locally varying widths, then by modeling the output distribution with potentially multimodal functions:
  - Heteroskedasticity ("local error bars"). Nix & Weigend (1995) described a method to train a network with two output units, the first giving the prediction, the second the error bar. Those two numbers, both functions of the input space, parametrize a Gaussian and can be used for unimodal densities. This method is more flexible than the constant variance assumption, but not appropriate for multimodal output densities.
  - The assumption of a single Gaussian can be generalized to a mixture of Gaussians that allow prediction of more general densities. Jacobs *et al.* (1991) introduced Gaussian mixture models to the connectionist community, and Weigend *et al.* (1995) applied them to time series prediction. As an alternative, rather than using this mixture of Gaussians with varying centers, Weigend & Srivastava (1995) introduced a fuzzy-logic like superposition of tent-functions at fixed centers to model potentially multimodal densities.
- Noisy inputs (observational noise). This important noise source in autoregression of noisy time series is well known in statistics and econometrics (see Section II-B) but less well known in the connectionist community (Weigend *et al.*, 1996). If the levels of the noise for each input is known, the effect can always be

[12]Other sources, important in nonlinear dynamical systems with low noise, such as the divergence of nearby trajectories, are less important in the present case of noisy financial data.

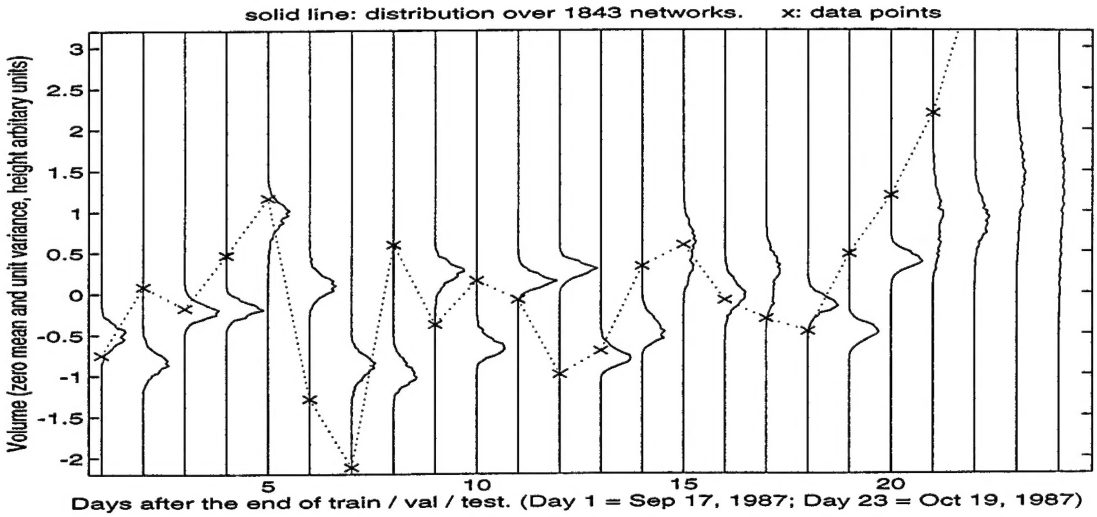

Fig. 6. One-day ahead densities for the days on the "super-test-set". The October 1987 stock market crash occurred on day 23 on this scale; the value for the volume went off scale.

emulated with a Monte Carlo simulation of forward passes with slightly different values for the inputs in order to build a density reflecting input fluctuations.

- Parameter noise (uncertainty in the weights). From a Bayesian perspective, model parameters are never known exactly but also have some uncertainty that translates into an uncertainty of the prediction (Buntine & Weigend, 1991; MacKay, 1992; Neal, 1996). While an analytic approximation is only available for simple cases, it is always possible to obtain a distribution by generating forward passes through networks with slightly different weight values.
- Regions of low input density (extrapolation). At the end of the Section III-D we discussed the uncertainty for the 1987 crash stemming from too few patterns in the vicinity of the pattern for which the prediction is to be made. For the data set used in this paper this is not an important source here since adjacent input patterns are highly correlated, implying that in most cases the network will have encountered nearby neighbors in the training set. This yields, however, to an overly optimistic interpretation of the performance.

The specific values of the prediction performance should not be overinterpreted. As typically done in cross-sectional bootstraps, we pick the validation and test patterns interspersed with the training data in order to obtain an indication of the variation of the subsample selection. Care has to be taken, however, in interpreting the results as accurate estimates of the generalization performance for truly future data. If there is a strong overlap from one pattern to the next (imagine a problem where all inputs are highly smoothed, like the exponentially filtered volatility estimate we use, or, even without smoothing, if the data is sampled with a frequency a lot faster than the dynamics of the system!) the chances are high that for a given test pattern, there will be very similar training patterns adjacent in time. In this case, more accurate estimates of the performance on future data might be obtained by bootstrapping blocks of data (Kunsch, 1989; Liu & Singh, 1992). Note, however, that these blocks are still taken from the entire period. So, if the dynamics is truly nonstationarity, this blocks-bootstrapping will still give overly optimistic results. To avoid fooling oneself on financial data, we strongly recommend using only data for testing that arrived after the end of the training and validation period (whether these two are interspersed, blocked, or sequential).

V. Conclusions

This article demonstrated the usefulness of a pairs bootstrap approach to generating and testing time series forecasts. We then applied the procedure to trading volume. Contrary to our expectation, no improvement over linear models could be obtained with a standard network trained with backpropagation and regularized by early stopping. This does not rule out the possibility of forecast improvements using additional forecast variables, or by using pruning and weight-elimination techniques.

The simulations gave us important insights into the variability of forecast performance over changes in subsamples and network structure. For our example, most of the variability in forecast performance was clearly coming from sample variation and not from model variation. This tells us that for this series there is probably little hope in fine tuning the networks we used. This is an example of an application where we feel that procedures such as bootstrapping are extremely useful in getting a clearer picture of what might be real and what is noise.

Acknowledgments

Blake LeBaron acknowledges support from the Alfred Sloan Foundation and the Wisconsin Alumni Research Foundation. Andreas Weigend acknowledges support from the National Science Foundation (ECS-9309786) and the Air Force Office of Scientific Research (F49620-96-1-0240), as well as the hospitality of the *Wirtschaftswissenschaftliche Fakultät der Humboldt Universität zu Berlin* during sum-

mer 1994. The computation of the time constant of the volatility persistence of this data set using a fully dynamic state space model is joint work with Jens Timmer, see Timmer & Weigend (1997).

REFERENCES

Bates, J. M. and C. W. J. Granger. 1969. The combination of forecasts. *Operations Research Quarterly* **20**, 451–468.

Bollerslev, T., R. Y. Chou, N. Jayaraman and K. F. Kroner. 1990. ARCH modeling in finance: A review of the theory and empirical evidence. *Journal of Econometrics* **52**(1), 5–60.

Bollerslev, T., R. F. Engle and D. B. Nelson. 1995. ARCH models. In *Handbook of Econometrics*, R. F. Engle and D. L. McFadden (eds), vol. 4, chapter 49. North-Holland, New York, NY.

Buntine, W. L. and A. S. Weigend. 1991. Bayesian backpropagation. *Complex Systems* **5**, 603–643.

Carroll, R., D. Ruppert and L. Stefanski. 1995. *Measurement Error in Nonlinear Models.* Chapman and Hall, London.

Connor, J. T. 1993. Bootstrap methods in neural network time series prediction. In *International Workshop on Applications of Neural Networks to Telecommunications*, J. Alspector, R. Goodman and T. X. Brown (eds), pp. 125–131, Hillsdale, NJ. Erlbaum.

Efron, B. and R. Tibshirani. 1993. *An Introduction to the Bootstrap.* Chapman and Hall, New York.

Efron, B. 1982. *The Jackknife, The Bootstrap, and Other Resampling Plans*, vol. 38 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, Philadelphia, Pennsylvania.

Fuller, W. A. 1987. *Measurement Error Models.* John Wiley, New York.

Gallant, A. R., P. E. Rossi and G. Tauchen. 1993. Nonlinear dynamic structures. *Econometrica* **61**, 871–908.

Jacobs, R. A. 1995. Methods for combining experts' probability assessments. *Neural Computation* **7**, 867–888.

Jacobs, R. A., M. I. Jordan, S. J. Nowlan and G. E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation* **3**, 79–87.

Karpov, J. M. 1987. The relation between price changes and trading volume: A survey. *Journal of Financial and Quantitative Analysis* **22**, 109–126.

Kunsch, H. R. 1989. The jacknife and the bootstrap for general stationary observations. *Annals of Statistics* **17**(3), 1217–1241.

LeBaron, B. 1992. Persistence of the Dow Jones index on rising volume. Technical report, University of Wisconsin - Madison, Madison, Wisconsin.

LeBaron, B. 1992. Some relations between volatility and serial correlations in stock market returns. *Journal of Business* **65**, 199–219.

Liu, R. Y. and K. Singh. 1992. Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the Limits of the Bootstrap*, R. LePage and L. Billard (eds), pp. 225–248. John Wiley, New York, NY.

MacKay, D. J. C. 1992. A practical Bayesian framework for backpropagation networks. *Neural Computation* **4**, 448–472.

Neal, R. 1996. *Bayesian Learning for Neural Networks.* Number 118 in Lecture Notes in Statistics. Springer-Verlag, New York.

Nix, D. A. and A. S. Weigend. 1995. Learning local error bars for nonlinear regression. In *Advances in Neural Information Processing Systems 7 (NIPS*94)*, G. Tesauro, D. S. Touretzky and T. K. Leen (eds), pp. 488–496. MIT Press, Cambridge, MA.

Paass, G. 1993. Assessing and improving neural network predictions by the bootstrap algorithm. In *Advances in Neural Information Processing Systems 5 (NIPS*92)*, S. J. Hanson, J. D. Cowan and C. L. Giles (eds), pp. 196–203, San Mateo, CA. Morgan Kaufmann.

Tibshirani, R. 1996. A comparison of some error estimates for neural network models. *Neural Computation* **8**, 152–163.

Timmer, J. and A. S. Weigend. 1997. Modeling volatility using state space models. *International Journal of Neural Systems* **8**, forthcoming.

Weigend, A. S. and A. N. Srivastava. 1995. Predicting conditional probability distributions: A connectionist approach. *International Journal of Neural Systems* **6**, 109–118.

Weigend, A. S., B. A. Huberman and D. E. Rumelhart. 1990. Predicting the future: A connectionist approach. *International Journal of Neural Systems* **1**, 193–209.

Weigend, A. S., B. A. Huberman and D. E. Rumelhart. 1992. Predicting sunspots and exchange rates with connectionist networks. In *Nonlinear Modeling and Forecasting*, M. Casdagli and S. Eubank (eds), pp. 395–432. Addison-Wesley.

Weigend, A. S., M. Mangeas and A. N. Srivastava. 1995. Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. *International Journal of Neural Systems* **6**, 373–399.

Weigend, A. S., H. G. Zimmermann and R. Neuneier. 1996. Clearning. In *Neural Networks in Financial Engineering (Proceedings of the Third International Conference on Neural Networks in the Capital Markets, NNCM-95)*, A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend (eds), pp. 511–522, Singapore. World Scientific.

Citation of this paper: LeBaron, B., and A. S. Weigend. 1998. A Bootstrap Evaluation of the Effect of Data Splitting on Financial Time Series. *IEEE Transactions on Neural Networks* **9**, forthcoming.
http://www.stern.nyu.edu/~aweigend/Research/Papers/Bootstrap